

RELAZIONE DI PROGETTO DI PERVASIVE COMPUTING

---

# Trasporti 116117 - Digital Twin

---

*Gibertoni Giada, Galassi Meshua*

# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Obiettivo del progetto</b>   | <b>3</b>  |
| <b>2</b> | <b>Analisi del dominio</b>  | <b>4</b>  |
| 2.1      | Il dominio . . . . .  | 4         |
| 2.1.1    | Normativa trasporti sanitari . . . . .                                | 5         |
| 2.1.2    | Situazione attuale AUSL della Romagna . . . . .                       | 5         |
| 2.1.3    | Proposta di riorganizzazione . . . . .                                | 6         |
| 2.2      | Incontro con l'esperto del dominio - 19.03.2021 . . . . .             | 7         |
| 2.3      | Scenari d'esempio . . . . .   | 9         |
| 2.3.1    | Scenario 1 - Trasporto da ospedale a casa . . . . .                   | 9         |
| 2.3.2    | Scenario 2 - Trasporto da casa all'ospedale . . . . .                 | 10        |
| 2.3.3    | Scenario 3 - Trasporto Trattamenti Ricorrenti Brevi . . . . .         | 11        |
| 2.3.4    | Scenario 4 - Trasporto Trattamenti Ricorrenti Lunghi . . . . .        | 11        |
| 2.4      | Requisiti . . . . .   | 12        |
| 2.5      | Considerazioni . . . . .  | 15        |
| 2.6      | Applicazione dell'approccio basato su Digital Twin . . . . .          | 16        |
| 2.6.1    | Physical asset . . . . .  | 16        |
| <b>3</b> | <b>Soluzione basata su Digital Twin</b>                               | <b>18</b> |
| 3.1      | Descrizione astratta del sistema . . . . .                            | 18        |
| 3.1.1    | Modelli di Digital Twins che rappresentano i physical asset . . . . . | 18        |
| 3.1.2    | Servizi e Funzionalità offerti dai Digital Twin . . . . .             | 31        |
| 3.1.3    | Architettura logica del sistema . . . . .                             | 33        |
| 3.1.4    | Lifecycle dei Digital Twins . . . . .                                 | 34        |
| 3.2      | Implementazione prototipale basata su Azure DT . . . . .              | 35        |
| 3.2.1    | Ambulance client . . . . .  | 36        |
| 3.2.2    | Call Center client . . . . .  | 36        |
| 3.2.3    | Simulazione dati di telemetria in arrivo dal GPS . . . . .            | 37        |
| 3.2.4    | Time Series Insight . . . . .   | 39        |
| 3.3      | Come avviene lo shadowing . . . . .                                   | 44        |
| 3.4      | Digital Twin API . . . . .  | 44        |
| 3.5      | Implementazione dei servizi forniti dai Digital Twins . . . . .       | 45        |

|          |  |           |
|----------|--|-----------|
| <b>4</b> | <b>Validazione e Test</b>  | <b>52</b> |
| 4.1      | Testing automatizzato . . . . .  | 52        |
| 4.2      | Implementazione Scenari . . . . .  | 52        |
| 4.2.1    | Scenario 1 . . . . .   | 52        |
| 4.2.2    | Altri scenari . . . . .  | 55        |
| <b>5</b> | <b>Discussione e conclusione</b>   | <b>56</b> |
| 5.1      | Conoscenze acquisite da questa sperimentazione . . . . .   | 56        |
| 5.1.1    | Azure Digital Twins . . . . .  | 56        |
| 5.2      | Efficacia dell'approccio basato su Digital Twin applicato al dominio<br>Trasporti 116117 . . . . . | 57        |
| 5.3      | Aspetti rilevanti non affrontati e miglioramenti futuri . . . . .                                  | 57        |
| 5.3.1    | Gestione delle ambulanze . . . . .   | 57        |
| 5.3.2    | Ottimizzazione dei percorsi . . . . .  | 57        |
| 5.3.3    | Analisi, simulazioni e predizioni . . . . .  | 58        |
| 5.3.4    | Archiviazione dei dati . . . . .   | 58        |

# 1 Obiettivo del progetto

Il progetto consiste nello studio e nell'implementazione di una serie opportuna di digital twins in un contesto di gestione in ambito healthcare. In particolare, si tratta di un progetto che mira ad un miglioramento della gestione dei trasporti ospedalieri secondari, ovvero, ciò di cui si occupa il numero unico europeo 116117 adibito alle cure mediche non urgenti ed altri servizi territoriali a bassa priorità. Tra tutti i servizi erogati attraverso questo numero, ci si focalizzerà prevalentemente su ciò che riguarda i trasporti ospedalieri non urgenti. Ad esempio, trasferimenti casa - ospedale e viceversa, oppure, trasporti tra strutture ospedaliere. Si è scelto di applicare il paradigma dei digital twins 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.

## 2 Analisi del dominio

### 2.1 Il dominio

Il servizio telefonico 116117 è un numero unico europeo che permette di accedere a cure mediche non urgenti ed altri servizi sanitari territoriali a bassa priorità. È un servizio che si rivolge a tutti i cittadini, sia italiani che stranieri, in maniera totalmente gratuita e fruibile da qualsiasi apparecchio, senza richiedere una registrazione preventiva. È un numero a chiamata rapida, non necessita di prefisso ed è disponibile 24 ore al giorno per 7 giorni alla settimana per fornire assistenza o informazioni. Attraverso il numero 116117 si gestiscono diversi settori:

1. Medico Continuità Assistenziale: è un servizio che permette di filtrare le chiamate ai medici per le visite domiciliari, alle USCA, alla guardia medica turistica ed alla centrale operativa 118.
2. 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.
3. Infermiere di territorio: ha l'obiettivo di fornire informazioni agli utenti e svolgere funzioni utili al monitoraggio ed all'assistenza in telemedicina.
4. 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.

In questo progetto, ci focalizzeremo sulla gestione della centrale dei trasporti secondari con lo scopo di rendere più efficiente il servizio fornito attualmente. In particolare, si vorrà rendere possibile un tracciamento dei mezzi di trasporto per migliorare la loro efficienza e sfruttamento ed evitare situazioni che portano ad uno spreco delle risorse disponibili. Inoltre, si vuole rendere possibile il tracciamento e il monitoraggio di tutti gli eventi rilevanti relativi ai trasporti secondari, così da produrre report e studiare l'andamento del servizio tramite indicatori statistici e KPI. Il sistema, potrebbe essere utile per costruire simulazioni di situazioni reali per aiutare la pianificazione e la gestione delle risorse.

### 2.1.1 Normativa trasporti sanitari

In materia di trasporti secondari esistono diverse normative di seguito riportate:

- **DM 70/2015** definisce gli standard qualitativi, strutturali e tecnologici da seguire per garantire una assistenza ospedaliera. Deve essere previsto lo sviluppo del servizio di emergenza territoriale tecnologicamente avanzato in modo tale da garantire una reale continuità dell'assistenza, anche attraverso la gestione tempestiva dei trasferimenti secondari urgenti in carico al 118 e la trasmissione di immagini e dati. Il DM sottolinea il fatto che sia nel caso in cui la gestione e il fabbisogno di mezzi dei trasporti secondari siano affidati a presidi ospedalieri, sia nel caso in cui siano affidati al 118 devono essere mantenuti separati dalla gestione del servizio di soccorso sanitario urgente.
- **DPCM 12 dicembre 2017** stabilisce che il sistema sanitario nazionale garantisca interventi sanitari tempestivi assicurando il trasporto in condizioni di sicurezza al presidio ospedaliero più appropriato.
- **Legge 17 luglio 2020, n. 77** stabilisce che, a causa dell'emergenza Covid-19, le regioni e le province autonome possano aumentare il numero di mezzi ed il relativo personale necessario per sostenere i trasporti secondari per pazienti Covid-19, per le dimissioni protette e per i trasporti interospedalieri di pazienti non affetti da Covid-19.
- **Regolamenti AUSL della Romagna** sottolinea che a causa dell'assenza di indicazioni regionali i trasporti secondari vengono gestiti in maniera diversa tra le varie aziende e anche all'interno di ognuna tra i diversi ambiti. Questo ha un effetto negativo poiché spesso in alcuni ambiti sono garantiti trasporti di un certo tipo e in altri no, questo varia il fabbisogno dei mezzi in maniera significativa.

### 2.1.2 Situazione attuale AUSL della Romagna

All'interno di AUSL della Romagna vi sono tre centrali operative per i trasporti secondari, di cui soltanto una è operativa h24. Queste centrali non sono collegate tra loro e 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.

Elementi di criticità:

- Assenza di un software unico tra le varie centrali con conseguenti difficoltà organizzative.
- Assenza di una rete radio unica con 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 nel tracciare i tempi degli stessi.

### 2.1.3 Proposta di riorganizzazione

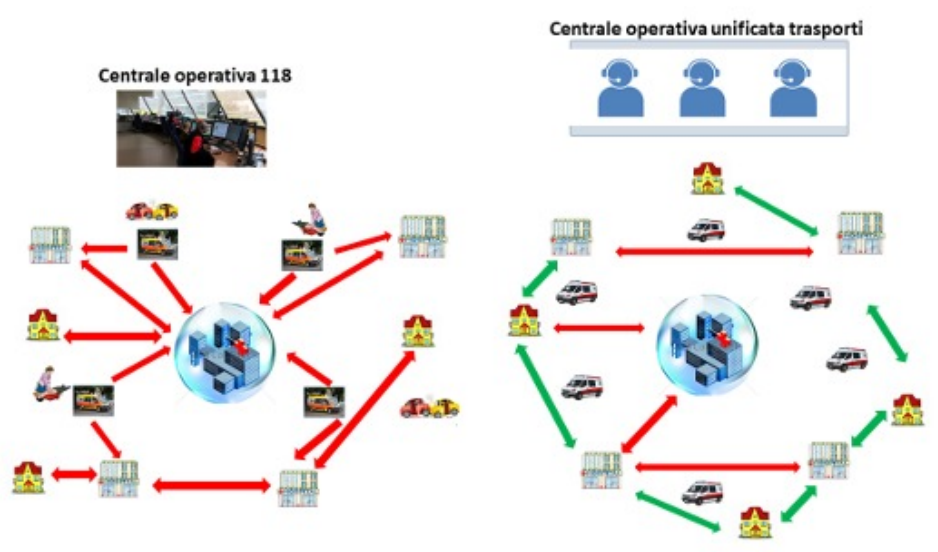


Figura 1: Modello ideale centrale operativa 118 e centrale operativa unificata trasporti

Seguendo anche quanto previsto dal DM 70/2015, si individuano due sistemi distinti:

- Sistema di emergenza sanitaria territoriale gestita dalla centrale operativa 118 che si occupa solamente dei servizi di emergenza (il suo lavoro si conclude, ove necessario, con il primo trasporto in ospedale).
- Sistema trasporti sanitari gestito dalla centrale unificata trasporti che si occupa solamente dei trasporti non in emergenza. Riguarda tutti i trasporti interospedalieri e gli altri trasporti da e verso il domicilio o altre strutture.

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 necessario un'organizzazione che tenga conto di entrambi i sistemi, per evitare inefficienze, sovrapposizioni e ritardi.

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 1.

## **2.2 Incontro con l'esperto del dominio - 19.03.2021**

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 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 116117 poiché non è stato progettato in precedenza.

Il 112 dovrebbe essere utilizzato per le emergenze e il 116117 per le non emergenze. Nel momento in cui le chiamate non sono indirizzate al giusto centralino, vengono 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 116117, 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 con ambulanze e le emergenze con ambulanze. 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. Si sono 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.

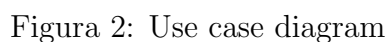
## **2.3 Scenari d'esempio**

### **2.3.1 Scenario 1 - Trasporto da ospedale a casa**

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 116117 per organizzare il trasporto. Durante la telefonata, l'operatore del centralino chiede la destinazione del paziente, l'orario delle dimissioni e le condizioni di mobilità del paziente. L'infermiere risponde che il paziente dovrà essere trasferito a Gambettola e 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. A questo punto l'operatore gli conferma che l'ambulanza è disponibile ed arriverà all'incirca verso le 16:00 e registra la prenotazione dell'ambulanza. Infine, l'operatore del centralino comunica, all'autista dell'ambulanza, la nuova prenotazione.

Alle 16:00, l'operatore del 116117 parte dalla propria sede (situata a Cesena) per raggiungere l'ospedale. Alle 16:15, l'operatore del 116177 arriva in reparto per



### 2.3.2 Scenario 2 - Trasporto da casa all'ospedale

In data 15 Aprile, il tutore del paziente effettua una chiamata al 116117 per prenotare il trasporto per il giorno 20 Maggio. Il centralino chiede gli indirizzi di partenza e destinazione, la data e l'orario in cui si dovrà svolgere il trasporto 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. Il ricovero è previsto per il giorno 20 Maggio a partire dalle ore 10:00. A questo punto, l'operatore del centralino conferma la prenotazione e la inserisce nel sistema.

Il giorno 20 Maggio, 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 reparto.

Il processo del trasporto si ritiene concluso e l'ambulanza può eseguire altri trasporti.

### **2.3.3 Scenario 3 - Trasporto Trattamenti Ricorrenti Brevi**

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 116117 e richiede il trasporto di andata e di ritorno del paziente, ogni giovedì alle 10:00, per tutta la durata del ricovero. Il centralino conferma le varie prenotazioni e le inserisce nel sistema.

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. L'operatore che si è occupato del trasporto 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 nel reparto in cui è ricoverato il trasporto si può considerare concluso.

### **2.3.4 Scenario 4 - Trasporto Trattamenti Ricorrenti Lunghi**

Un paziente, con insufficienza renale, ha necessità di seguire un trattamento di dialisi presso il reparto di nefrologia e dialisi dell'ospedale Bufalini di Cesena. Il trattamento è previsto per 3 sedute settimanali della durata di 4 ore.

Il paziente dovrà essere trasportato dal proprio domicilio all'ospedale. Alla prescrizione della terapia, il paziente contatta il 116117 e richiede il trasporto di andata e di ritorno per ogni lunedì, mercoledì e venerdì alle ore 8:00. Il centralino conferma le prenotazioni e le inserisce nel sistema.

Considerando la lunga durata del trattamento, l'operatore si occuperà di prenotare sia il mezzo che consentirà il trasporto dal domicilio del paziente al centro di dialisi, sia il mezzo che si occuperà del tragitto di ritorno.

Il trasporto di andata e quello di ritorno sono considerati due processi separati in quanto la durata del trattamento ha una lunghezza significativa che richiederebbe di tenere occupati gli operatori troppo a lungo. Ogni processo si ritiene completato all'arrivo alla destinazione (domicilio o ospedale).

## 2.4 Requisiti

Di seguito sono riportati i requisiti del nuovo sistema evidenziati dall'esperto di dominio.

- Il servizio 116117 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.
- **E' auspicabile l'integrazione di soluzioni tecnologiche tra 118, 112 NUE e NE 116117** 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.

- **Il corretto utilizzo dei sistemi di gestione automatica.** Nel centro di risposta NE 116117 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.
- **La gestione 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.
- **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 116117** 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.
- **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 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 dell'operatore dell'ambulanza.**
- **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 dei dati da parte di sistemi esterni.**

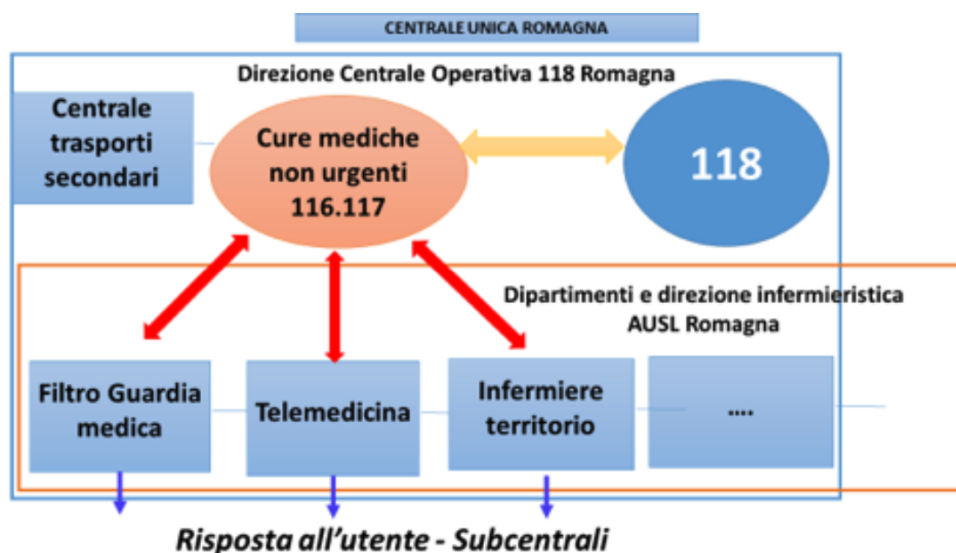


Figura 3: Organizzazione interna 116117

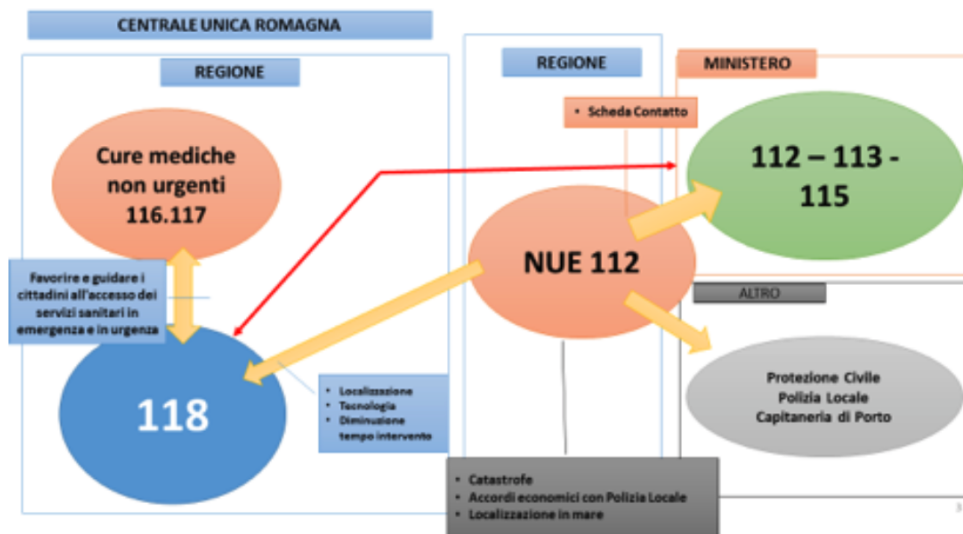


Figura 4: Comunicazione 116117, 118 e 112

## 2.5 Considerazioni

- Si tiene traccia delle chiamate ricevute registrando i dati relativi alla chiamata e alla relativa richiesta anche nel caso in cui la prenotazione non andasse a buon fine, così da poter fare delle analisi sul servizio.
- La registrazione delle varie prenotazioni verrà segnata dall'operatore del centralino come nuova attività prevista nella giornata e sarà l'operatore dell'ambulanza a consultare attivamente ogni giorno le attività indicate.
- Il paziente, inteso come persona trasportata, non agisce attivamente all'interno del sistema ma, resta solamente un'entità passiva che subisce le azioni degli attori. Il paziente che agisce attivamente nel sistema (richiede il servizio) è incluso all'interno dell'attore "Utente".
- Per ottimizzare l'utilizzo delle risorse disponibili, diversamente da quanto previsto nello scenario 3, si considerano gli eventi di andata e ritorno di un trasporto per trattamenti ricorrenti brevi, come due eventi distinti (come avviene nello scenario 4). In questo modo l'operatore non dovrà attendere la durata del trattamento e si potrà considerare libero per un'altra eventuale attività di trasporto.



- **La tipologia delle informazioni e dei servizi.** Nell'elenco dei requisiti sottolineati dall'esperto del dominio emerge la necessità di avere distinzione delle figure all'interno del centralino per gestire i vari servizi offerti dal 116117. Nel nostro modello non sarà tenuto in considerazione questo aspetto poiché si fa riferimento solamente al servizio dei trasporti.

## 2.6 Applicazione dell'approccio basato su Digital Twin

Si è scelto di seguire un approccio basato su digital twin poiché, questo ambito di gestione si presta particolarmente bene all'utilizzo di sensori per la rilevazione dei dati.

In particolare, si può facilmente rilevare la posizione dell'ambulanza, attraverso i dati derivanti dal GPS integrato in essa. Questo potrebbe facilitarne la gestione anche in merito ad una riduzione dei viaggi a vuoto che attualmente svolgono le ambulanze. Potendo rilevare che un'ambulanza, che ha già concluso il trasporto di cui si stava occupando, si trova in un luogo non molto lontano da un nuovo trasporto richiesto, si può scegliere di impiegare la stessa ambulanza, evitando di farne partire una seconda per recarsi nel medesimo luogo facendo, poi, ritornare la prima alla sede principale.

Si potrà, quindi, ottenere una migliore efficienza del servizio fornito evitando situazioni che portano allo spreco delle risorse disponibili.

Inoltre, un ulteriore aspetto per cui si è scelto di utilizzare un approccio basato su digital twin, è quello di permettere il tracciamento ed il monitoraggio di tutti gli eventi rilevanti relativi ai trasporti secondari, per poter, successivamente, produrre report e studiare l'andamento del servizio attraverso degli indicatori statistici.

Infine, proprio con l'obiettivo di migliorare la gestione, il sistema potrebbe essere utile nelle simulazioni di situazioni reali per sostenere le operazioni di pianificazione e verificare, preventivamente, gli effetti che potrebbe avere una modifica che si intende apportare alla gestione o ai mezzi messi a disposizione.

### 2.6.1 Physical asset

- **Trasporto:** asset centrale del sistema. Rappresenta il processo di trasporto che comincia dalla presa in carico del paziente e termina con il suo arrivo a destinazione.

- **Ambulanza e GPS:** mezzo di trasporto guidato dall'operatore per eseguire la richiesta del servizio. All'interno di ogni ambulanza è presente un sensore GPS.
- **Operatore ambulanza:** persona che svolge il servizio di trasporto.
- **Paziente:** persona che usufruisce del servizio di trasporto.

Alla ricezione della chiamata il centralino potrà rispondere ad una richiesta di informazioni oppure ad una richiesta di servizio.

- **Richiesta di servizio:** richiesta effettiva di prenotazione per un trasporto. La richiesta è rappresentata dalla data e dall'ora di ricezione della chiamata. Durante questa chiamata viene richiesto un trasporto da parte dell'utente, se la richiesta dell'utente è realizzabile viene registrata la prenotazione, altrimenti la richiesta di servizio sarà conclusa.
- **Prenotazione:** durante una richiesta di servizio, se possibile, viene registrata la prenotazione del trasporto. Quest'ultima è rappresentata dall'itinerario (luogo di partenza e luogo di destinazione), dalla data e ora in cui dovrà avvenire il trasporto, dall'ambulanza che effettuerà il trasporto ed infine dai dati del paziente da trasportare.
- **Richiesta di informazioni:** richiesta nel quale l'utente chiama il numero unico 116117 solamente a scopo informativo e non per effettuare prenotazioni.

## 3 Soluzione basata su Digital Twin

### 3.1 Descrizione astratta del sistema

#### 3.1.1 Modelli di Digital Twins che rappresentano i physical asset

I physical asset riportati sopra sono stati rappresentati digitalmente mediante i relativi digital twin. Degli asset fisici sono state catturate solo le proprietà di interesse per il dominio, come di seguito riportato.

Per la rappresentazione come digital twin degli asset fisici prenotazione, richiesta di servizio e richiesta di informazioni si è discusso molto, in quanto, in alternativa, potevano essere semplici dati memorizzabili in una database. Si è giunti alla conclusione che questi digital twin potrebbero essere molto utili nelle fasi di analisi del sistema 116117. Questo è un punto molto importante emerso durante le interviste con l'esperto del dominio. Per questo motivo si è scelto di digitalizzare anche questi tre physical asset.

- **Trasporto:** è costituito da data ed ora effettiva di inizio e fine trasporto. In questo modo si possono, al bisogno, analizzare discostamenti dell'orario (anticipi o ritardi) rispetto all'orario di prenotazione del trasporto.

Inoltre, al suo interno viene mantenuto il relativo itinerario.

Questo digital twin sarà relazionato con il paziente da trasportare, l'ambulanza interessata e l'operatore che svolge il trasporto. Sarà anche collegato alla relativa prenotazione.

Di seguito è riportato il suo modello:

---

```
{"@id": "dtmi:num116117:transport;1",
"@type": "Interface",
"@context": ["dtmi:dtdl:context;2"],
"displayName": "Transport",
"contents": [
  {"@type": "Property",
   "name": "route",
   "schema": "dtmi:num116117:transportRoute;1"},
  {"@type": "Property",
   "name": "startDateTime",
```

```

    "schema": "dateTime"},
{"@type": "Property",
  "name": "endTime",
  "schema": "dateTime"},
{"@type": "Relationship",
  "name": "use"},
{"@type": "Relationship",
  "name": "transport"},
{"@type": "Relationship",
  "name": "drive"},
{"@type": "Relationship",
  "name": "related"}
],
"schemas": [
{"@id": "dtmi:num116117:transportRoute;1",
  "@type": "Object",
  "fields": [
    {"name": "departure",
      "schema": "dtmi:num116117:transportLocation;1"},
    {"name": "destination",
      "schema": "dtmi:num116117:transportLocation;1"}
  ]
},
{"@id": "dtmi:num116117:transportLocation;1",
  "@type": "Object",
  "fields": [
    {"name": "address",
      "schema": "dtmi:num116117:transportAddress;1"},
    {"name": "houseNumber",
      "schema": "dtmi:num116117:transportHouseNumber;1"},
    {"name": "city",
      "schema": "dtmi:num116117:transportCity;1"},
    {"name": "district",
      "schema": "dtmi:num116117:transportDistrict;1"},
    {"name": "postalCode",
      "schema": "dtmi:num116117:transportPostalCode;1"}
  ]
}

```

```

    ]
  },
  {"@id": "dtmi:num116117:transportAddress;1",
    "@type": "Object",
    "fields": [{
      "name": "address",
      "schema": "string"}]
  },
  {"@id": "dtmi:num116117:transportHouseNumber;1",
    "@type": "Object",
    "fields": [{
      "name": "houseNumber",
      "schema": "integer"}]
  },
  {"@id": "dtmi:num116117:transportCity;1",
    "@type": "Object",
    "fields": [{
      "name": "city",
      "schema": "string"}]
  },
  {"@id": "dtmi:num116117:transportPostalCode;1",
    "@type": "Object",
    "fields": [{
      "name": "postalCode",
      "schema": "integer"}]
  },
  {"@id": "dtmi:num116117:transportDistrict;1",
    "@type": "Object",
    "fields": [{
      "name": "district",
      "schema": "string"}]
  }
]
}

```

---

- **Ambulanza e GPS:** rappresenta il veicolo fisico ed, al momento, è costituita

da un identificativo ed è collegata ad un GPS che permette di tracciare i suoi spostamenti e monitorare, quindi, in tempo reale la sua posizione.

In un futuro sviluppo il digital twin dell'ambulanza potrebbe essere integrato facilmente con ulteriori dati e sensori.

Di seguito è riportato il suo modello:

---

```
{"@id": "dtmi:num116117:ambulance;1",
  "@type": "Interface",
  "@context": [
    "dtmi:dtdl:context;2"
  ],
  "displayName": "Ambulance",
  "contents": [
    {"@type": "Property",
      "name": "number",
      "schema": "integer"},
    {"@type": "Relationship",
      "name": "use"},
    {"@type": "Relationship",
      "name": "contains"}
  ]}
```

```
{"@id": "dtmi:num116117:GPS;1",
  "@type": "Interface",
  "@context": [
    "dtmi:dtdl:context;2"
  ],
  "displayName": "GPS",
  "contents": [
    {"@type": "Property",
      "name": "longitude",
      "schema": "double"},
    {"@type": "Property",
      "name": "latitude",
      "schema": "double"},
    {"@type": "Relationship",
```

```
    "name": "contains"}
  ]}
```

---

- **Operatore ambulanza:** rappresenta la persona che effettua i trasporti a bordo dell'ambulanza. Per quanto emerso dall'analisi del dominio, questo digital twin è rappresentato solamente dai suoi dati personali e dalla relazione con il trasporto. Quest'ultimo punto può essere utile in fase di analisi o nel caso in cui sorgessero problemi relativi ad un determinato trasporto, in quanto il trasporto sarebbe direttamente riconducibile all'operatore che l'ha effettuato.

Di seguito è riportato il suo modello:

---

```
{ "@id": "dtmi:num116117:ambulanceOperator;1",
  "@type": "Interface",
  "@context": [
    "dtmi:dtdl:context;2"
  ],
  "displayName": "AmbulanceOperator",
  "contents": [
    { "@type": "Property",
      "name": "personalData",
      "schema": "dtmi:num116117:operatorPersonalData;1" },
    { "@type": "Relationship",
      "name": "drive" }
  ],
  "schemas": [
    { "@id": "dtmi:num116117:operatorPersonalData;1",
      "@type": "Object",
      "fields": [
        { "name": "name",
          "schema": "string" },
        { "name": "surname",
          "schema": "string" },
        { "name": "birthDate",
          "schema": "date" },
```

```

    {"name": "residence",
     "schema": "dtmi:num116117:operatorResidence;1"}
  ]},
{"@id": "dtmi:num116117:operatorResidence;1",
 "@type": "Object",
 "fields": [
  {"name": "address",
   "schema": "dtmi:num116117:operatorAddress;1"},
  {"name": "houseNumber",
   "schema": "dtmi:num116117:operatorHouseNumber;1"},
  {"name": "city",
   "schema": "dtmi:num116117:operatorCity;1"},
  {"name": "district",
   "schema": "dtmi:num116117:operatorDistrict;1"},
  {"name": "postalCode",
   "schema": "dtmi:num116117:operatorPostalCode;1"}
 ]
},
{"@id": "dtmi:num116117:operatorAddress;1",
 "@type": "Object",
 "fields": [
  {"name": "address",
   "schema": "string"}
 ]},
{"@id": "dtmi:num116117:operatorHouseNumber;1",
 "@type": "Object",
 "fields": [
  {"name": "houseNumber",
   "schema": "integer"}
 ]
},
{"@id": "dtmi:num116117:operatorCity;1",
 "@type": "Object",
 "fields": [
  {"name": "city",
   "schema": "string"}
 ]
}

```



```

    ]
  },
  {"@id": "dtmi:num116117:operatorDistrict;1",
    "@type": "Object",
    "fields": [
      {"name": "district",
        "schema": "string"}
    ]
  },
  {"@id": "dtmi:num116117:operatorPostalCode;1",
    "@type": "Object",
    "fields": [
      {"name": "postalCode",
        "schema": "integer"}
    ]
  }
]}

```

---

- **Paziente:** rappresenta digitalmente la persona che usufruisce del servizio, vengono mantenuti i suoi dati anagrafici. Inoltre, incapsula anche lo stato di salute del paziente e la sua autonomia in modo da permettere all'operatore che effettua il trasporto di conoscere lo stato fisico e cognitivo del paziente prima di avere contatti con lui. Questo permette una migliore organizzazione e resa del servizio.

Al momento non ci sono ulteriori sensori a bordo dell'ambulanza che possano descrivere la salute del paziente. Nel caso in cui un giorno venissero aggiunti nell'asset fisico, sarebbe semplice integrarle nel relativo digital twin.

Per concludere, il paziente è relazionato con il trasporto e con la prenotazione. Questo, come nel caso precedente, è utile per una fase di analisi dei servizi offerti ed anche per risalire ai vari spostamenti in caso di eventuali problemi.

Di seguito è riportato il suo modello:

---

```

{"@id": "dtmi:num116117:patient;1",
  "@type": "Interface",
  "@context": ["dtmi:dtdl:context;2"],
  "displayName": "Patient",

```

```

"contents": [
  {"@type": "Property",
   "name": "autonomy",
   "schema": "dtmi:num116117:patientAutonomy;1"},
  {"@type": "Property",
   "name": "personalData",
   "schema": "dtmi:num116117:patientPersonalData;1"},
  {"@type": "Property",
   "name": "healthState",
   "schema": "dtmi:num116117:healthState;1"},
  {"@type": "Relationship",
   "name": "transport"}
],
"schemas": [
  {"@id": "dtmi:num116117:patientPersonalData;1",
   "@type": "Object",
   "fields": [
     {"name": "name",
      "schema": "string"},
     {"name": "surname",
      "schema": "string"},
     {"name": "birthDate",
      "schema": "date"},
     {"name": "residence",
      "schema": "dtmi:num116117:patientResidence;1"}
   ]
  },
  {"@id": "dtmi:num116117:healthState;1",
   "@type": "Object",
   "fields": [
     {"name": "description",
      "schema": "string"}
   ]
  },
  {"@id": "dtmi:num116117:patientAutonomy;1",
   "@type": "Enum",

```

```

"valueSchema": "integer",
"enumValues": [
  {"name": "autonomous",
   "displayName": "Autonomous",
   "enumValue": 1},
  {"name": "partiallyAutonomous",
   "displayName": "PartiallyAutonomous",
   "enumValue": 2},
  {"name": "notAutonomous",
   "displayName": "NotAutonomous",
   "enumValue": 3}
]
},
{
  "@id": "dtmi:num116117:patientResidence;1",
  "@type": "Object",
  "fields": [
    {"name": "address",
     "schema": "dtmi:num116117:patientAddress;1"},
    {"name": "houseNumber",
     "schema": "dtmi:num116117:patientHouseNumber;1"},
    {"name": "city",
     "schema": "dtmi:num116117:patientCity;1"},
    {"name": "district",
     "schema": "dtmi:num116117:patientDistrict;1"},
    {"name": "postalCode",
     "schema": "dtmi:num116117:patientPostalCode;1"}
  ]
},
{"@id": "dtmi:num116117:patientAddress;1",
 "@type": "Object",
 "fields": [
  {"name": "address",
   "schema": "string"}
 ]},
{"@id": "dtmi:num116117:patientHouseNumber;1",

```

```

"@type": "Object",
"fields": [
  {"name": "houseNumber",
   "schema": "integer"}
]],
{"@id": "dtmi:num116117:patientCity;1",
 "@type": "Object",
"fields": [
  {"name": "city",
   "schema": "string"}
]],
{"@id": "dtmi:num116117:patientDistrict;1",
 "@type": "Object",
"fields": [
  {"name": "district",
   "schema": "string"}
]],
{"@id": "dtmi:num116117:patientPostalCode;1",
 "@type": "Object",
"fields": [
  {"name": "postalCode",
   "schema": "integer"}
]]
}}

```

---

- **Prenotazione:** rappresenta la prenotazione di un trasporto, descrive l'itinerario del trasporto da effettuare e la relativa data e ora.

Di seguito è riportato il suo modello:

---

```

{"@id": "dtmi:num116117:booking;1",
 "@type": "Interface",
"@context": [
  "dtmi:dtdl:context;2"
],
"displayname": "Booking",
"contents": [

```

```

{"@type": "Property",
  "name": "route",
  "schema": "dtmi:num116117:route;1"},
{"@type": "Property",
  "name": "dateTime",
  "schema": "dateTime"},
{"@type": "Relationship",
  "name": "transport"},
{"@type": "Relationship",
  "name": "requestBy"},
{"@type": "Relationship",
  "name": "related"}
],
"schemas": [
{"@id": "dtmi:num116117:route;1",
  "@type": "Object",
  "fields": [
    {"name": "departure",
      "schema": "dtmi:num116117:location;1"},
    {"name": "destination",
      "schema": "dtmi:num116117:location;1"}
  ]},
{"@id": "dtmi:num116117:location;1",
  "@type": "Object",
  "fields": [
    {"name": "address",
      "schema": "dtmi:num116117:address;1"},
    {"name": "houseNumber",
      "schema": "dtmi:num116117:houseNumber;1"},
    {"name": "city",
      "schema": "dtmi:num116117:city;1"},
    {"name": "district",
      "schema": "dtmi:num116117:district;1"},
    {"name": "postalCode",
      "schema": "dtmi:num116117:postalCode;1"}
  ]},

```

```

{"@id": "dtmi:num116117:address;1",
  "@type": "Object",
  "fields": [
    {"name": "address",
      "schema": "string"}
  ]},
{"@id": "dtmi:num116117:houseNumber;1",
  "@type": "Object",
  "fields": [
    {"name": "houseNumber",
      "schema": "integer"}
  ]},
{"@id": "dtmi:num116117:city;1",
  "@type": "Object",
  "fields": [
    {"name": "city",
      "schema": "string"}
  ]},
{"@id": "dtmi:num116117:postalCode;1",
  "@type": "Object",
  "fields": [
    {"name": "postalCode",
      "schema": "integer"}
  ]},
{"@id": "dtmi:num116117:district;1",
  "@type": "Object",
  "fields": [
    {"name": "district",
      "schema": "string"}
  ]}
]}

```

---

- **Richiesta di servizio:** è separata dalla prenotazione in quanto si vuole tenere traccia delle richieste di servizio che non sono andate a buon fine, ovvero che non hanno portato ad una prenotazione. Ciò può essere utile per

analizzare il numero di richieste di servizio che si ricevono a confronto con quelle che si riescono poi effettivamente a soddisfare. In questo modo si può capire se le risorse disponibili sono sufficienti per il livello di richieste oppure se devono essere integrate.

Di seguito è riportato il suo modello:

---

```
{
  "@id": "dtmi:num116117:serviceRequest;1",
  "@type": "Interface",
  "@context": [
    "dtmi:dtdl:context;2"
  ],
  "displayName": "ServiceRequest",
  "contents": [
    {"@type": "Property",
      "name": "serviceRequestId",
      "schema": "string"},
    {"@type": "Property",
      "name": "dateTime",
      "schema": "dateTime"},
    {"@type": "Relationship",
      "name": "requestBy"}
  ]
}
```

---

- **Richiesta di informazioni:** è una semplice richiesta di informazioni. Tiene traccia, oltre che della data e dell'ora di ricezione della richiesta, delle informazioni richieste durante la chiamata. Anche questo digital twin è molto utile a livello di analisi.

Di seguito è riportato il suo modello:

---

```
{"@id": "dtmi:num116117:infoRequest;1",
  "@type": "Interface",
  "@context": [
    "dtmi:dtdl:context;2"
  ],
  "displayName": "InfoRequest",
```

```
"contents": [  
  {"@type": "Property",  
    "name": "infoRequestId",  
    "schema": "string"},  
  {"@type": "Property",  
    "name": "dateTime",  
    "schema": "dateTime"},  
  {"@type": "Property",  
    "name": "description",  
    "schema": "string"}  
]
```

---

### 3.1.2 Servizi e Funzionalità offerti dai Digital Twin

Dagli incontri con l'esperto del dominio sono stati evidenziati vari servizi utili che i digital twin dovrebbero fornire:

- **Organizzazione dei mezzi di trasporto** Uno dei problemi principali emersi durante il knowledge crunching è la disorganizzazione attuale dei mezzi di trasporto e il conseguente spreco di risorse.

Tramite i digital twin, in particolare quelli riguardanti il bounded context relativo al processo di trasporto, è possibile effettuare un'organizzazione più accurata in quanto mettono a disposizione una serie di funzionalità che permettono un corretto utilizzo delle risorse a disposizione.

Tramite il digital twin ambulanza è possibile sapere in tempo reale l'esatta collocazione delle varie ambulanze presenti all'interno dell'organizzazione. Inoltre il digital twin relativo al trasporto permette di vedere tutti i trasporti in corso sul territorio.

Grazie a questi due digital twin, si può integrare il sistema con un servizio che va ad ottimizzare le assegnazioni delle ambulanze ai vari trasporti, in modo da evitare sprechi di risorse.

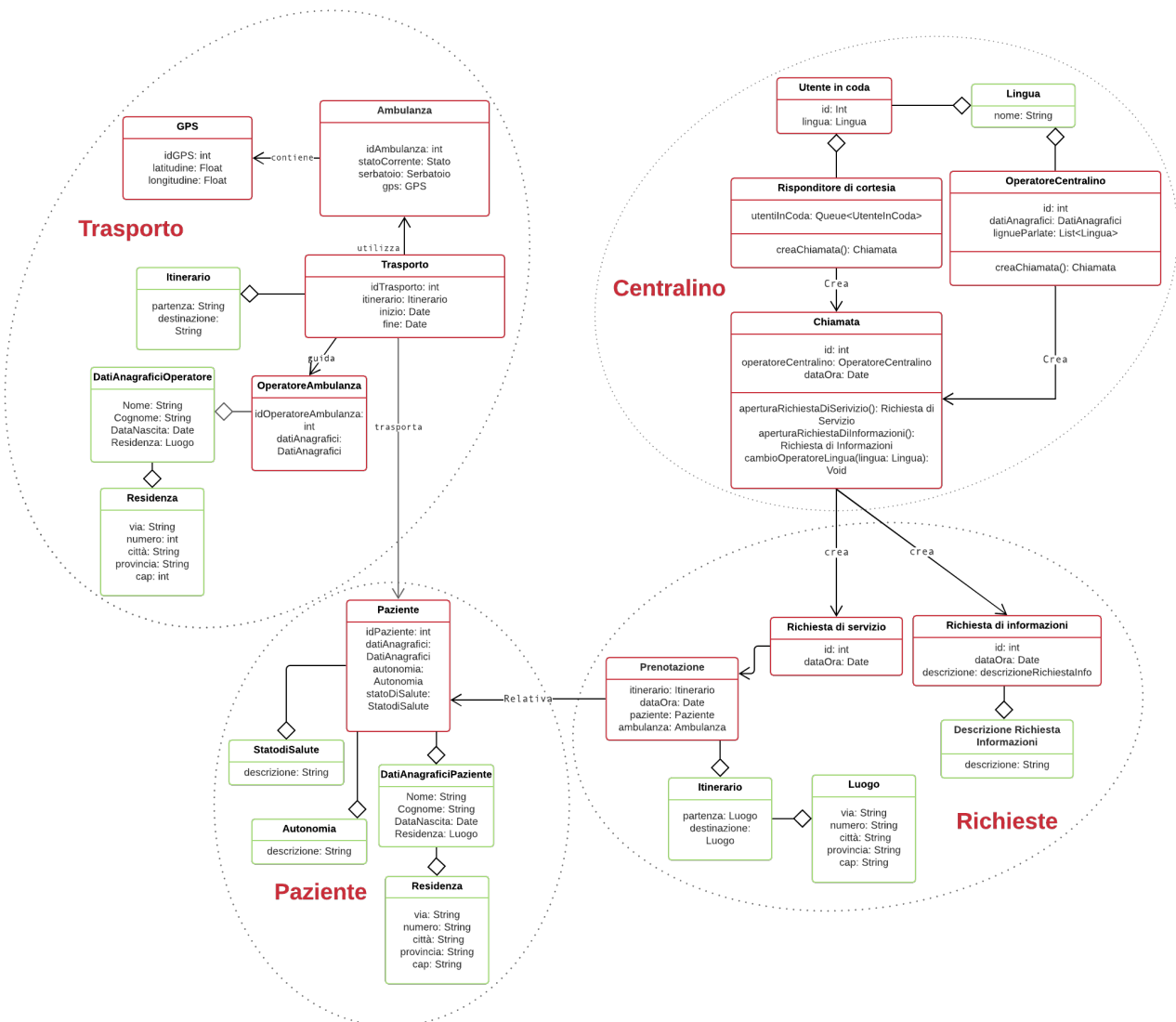
- **Monitoraggio del sistema** Un altro problema emerso è la disorganizzazione delle informazioni. Il sistema ora permette di tracciare tutte le informazioni rilevanti, dalla chiamata per la prenotazione del trasporto fino alla



conclusione dello stesso. In questo modo, in caso di problemi è possibile risalire a qualsiasi informazione riguardante il processo. Il digital twin del paziente è legato a tutte le prenotazioni e a tutti i trasporti di cui ha usufruito. Dunque è possibile monitorare tutti i suoi spostamenti, anche in tempo reale.

- **Analisi del sistema** La struttura costruita permette di analizzare il sistema dalla chiamata dell'utente fino alla conclusione del trasporto. Come richiesto dall'esperto di dominio, il sistema permetterà un'analisi approfondita per risolvere determinati problemi come, ad esempio, analizzare il flusso di richieste di servizio che non sono andate a buon fine per mancanza di disponibilità e quindi integrare personale o mezzi ove necessario.
- **Comunicazione con sistemi esterni eterogenei** Uno dei requisiti del sistema era l'interoperabilità con sistemi esterni. Grazie alla piattaforma utilizzata per costruire e gestire i digital twin questo requisito è facilmente raggiungibile, in quanto la piattaforma mette a disposizione svariate REST API che promuovono l'interoperabilità con molti sistemi.

### 3.1.3 Architettura logica del sistema



Come si può vedere in figura, abbiamo suddiviso il sistema in quattro bounded context che rappresentano:

- **Centralino:** parte del sistema 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 caso in cui si tratti di un utente che parla solamente una lingua specifica e l'eventuale attesa in coda nel caso in cui tutti gli operatori del centralino siano occupati.

- **Richieste:** parte del sistema relativa a tutte le richieste effettuate dagli utenti al centralino. In questo ambito sono incluse sia le richieste fatte ad unico scopo informativo, sia le richieste di poter usufruire del servizio. Da queste ultime, eventualmente, verrà poi effettuata la vera e propria prenotazione del trasporto.
- **Paziente:** bounded context in cui sono state racchiuse tutte le informazioni relative al paziente, compreso il suo stato di salute e il grado di autonomia, utile da conoscere per organizzare il trasporto.
- **Trasporto:** parte centrale del sistema che riguarda tutto il processo di trasporto. Include, infatti, l'itinerario da seguire, l'ambulanza utilizzata e l'operatore che si occupa dello specifico servizio.

In un primo approccio, all'interno del sistema sono state implementate solamente le funzionalità relative ai due bounded context centrali ovvero, *Paziente* e *Trasporto*. Da una successiva analisi, è emersa, come riportato nel sottocapitolo 3.1.1, la necessità di implementare anche il bounded context relativo alle *Richieste*. Si è scelto, invece, di non implementare il bounded context relativo al *Centralino*, in quanto prevede un semplice salvataggio di dati gestibile attraverso un database.

### 3.1.4 Lifecycle dei Digital Twins

Al fine di effettuare successive analisi relative allo svolgimento del servizio, i digital twins, anche una volta terminato il processo fisico a cui sono associati, non verranno eliminati, ma verranno settati in uno stato concluso.

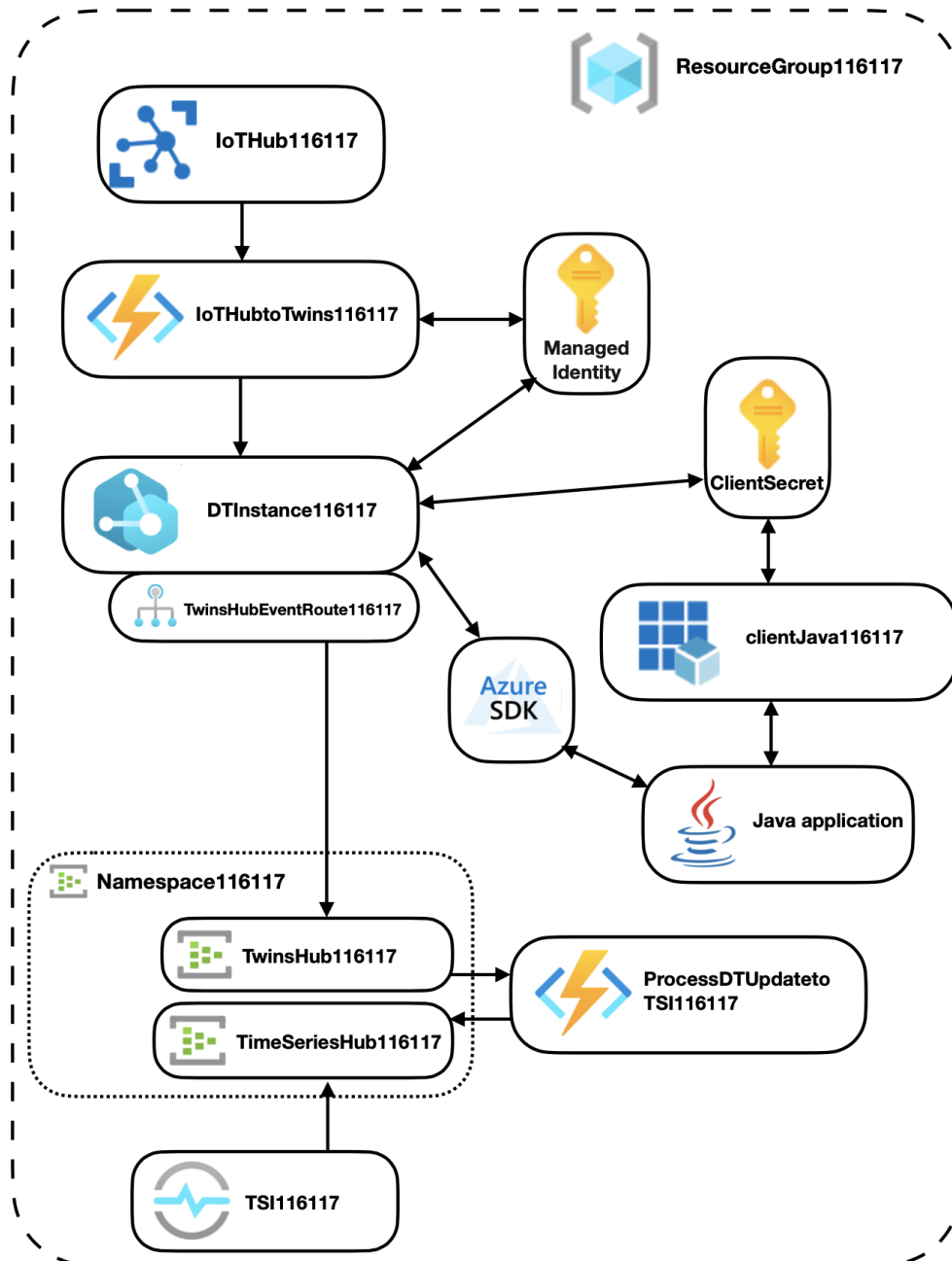
Considerando che si tratta di una rappresentazione digitale della realtà, una volta terminato il ciclo di vita dell'asset fisico, potrebbe essere opportuno eliminare anche il corrispondente elemento digitale.

Dopo un'attenta analisi si è ritenuto più conveniente non far corrispondere il lifecycle dell'asset fisico a quello del relativo gemello digitale, in quanto questo impedirebbe di effettuare un'analisi a posteriori dei servizi svolti.

Si ritiene opportuno, però, dopo un lasso di tempo prestabilito, terminare il lifecycle della rappresentazione digitale, ed, eventualmente memorizzare i relativi dati in un database esterno.

Il lasso di tempo dopo il quale eliminare i digital twin, si stima in un periodo oltre il quale le analisi svolte su di essi non rappresentano più situazioni veritiere perché ormai obsolete.

### 3.2 Implementazione prototipale basata su Azure DT



Per implementare il sistema è stato utilizzato Microsoft Azure.

Sono stati costruiti due client in Java sfruttando le API messe a disposizione da Azure. Come si può vedere in figura per creare, eliminare ed interrogare i digital twins all'interno di *DTInstance116117* è stato utilizzato il Java SDK messo a disposizione da Azure.

Per potere accedere alla digital twin instance, però, è necessario autenticarsi. Azure Identity mette a disposizione vari modi per farlo. In questo caso è stata registrata un'applicazione client in Azure Active Directory, una piattaforma per la gestione e la protezione delle identità. Per connettere l'app sul cloud a quella in locale si è utilizzato il `clientSecret`, ovvero una stringa usata dall'applicazione per dimostrare la propria identità durante la richiesta di un token. Dopodiché è stato assegnato il ruolo di "Azure Digital Twins Data Owner" nella *DTInstance116117* all'applicazione creata in Azure Active Directory, permettendogli di accedere all'istanza.

Di seguito una spiegazione più dettagliata dei client costruiti.

### **3.2.1 Ambulance client**

Questo è ipoteticamente il client presente a bordo dell'ambulanza. All'avvio dell'applicazione viene chiesto di identificare l'ambulanza e il relativo operatore. Successivamente vengono elencate le prenotazioni della giornata che non sono ancora state svolte. Da qui l'operatore potrà selezionare la prenotazione che intende svolgere. Alla selezione della prenotazione quindi parte effettivamente il trasporto. Dunque viene creato il rispettivo digital twin con data e ora della partenza effettiva. Fino al termine di questo trasporto, l'operatore non potrà iniziare nessun altro trasporto. Quando l'ambulanza giungerà a destinazione l'operatore potrà terminare il trasporto e il relativo digital twin verrà aggiornato, entrando in uno stato concluso. Dopodiché, l'operatore potrà continuare eseguendo un nuovo trasporto.

### **3.2.2 Call Center client**

Questo è il client utilizzato ipoteticamente dall'ufficio che gestisce le chiamate, le prenotazioni ed il sistema 116117 in generale. Tramite questa applicazione si potrà quindi:

- Registrare l'inserimento nel sistema di una nuova ambulanza oppure rimuovere un ambulanza in caso venga dismessa: consiste nella creazione e nell'eliminazione del digital twin relativo all'ambulanza.
- Inserire nel sistema un nuovo paziente: consiste nella creazione del digital twin relativo al paziente.
- Inserire nel sistema un nuovo operatore: consiste nella creazione del digital twin relativo all'operatore.
- Aprire una richiesta di servizio con la relativa prenotazione: consiste nella creazione del digital twin relativo alla richiesta di servizio e, in caso fosse possibile, della prenotazione.
- Aprire una richiesta di informazioni: consiste nella creazione del digital twin relativo alla richiesta di informazioni.
- Eliminare una prenotazione: consiste nell'eliminazione del digital twin relativo ad una determinata prenotazione.
- Tracciare una specifica ambulanza in tempo reale: data una certa ambulanza, consente di tracciare la sua posizione in tempo reale, interrogando il relativo digital twin.
- Visualizzare i trasporti in corso: visualizza tutti i digital twin corrispondenti ad un trasporto in corso.

Si è scelto di implementare solo alcune funzionalità per testare l'utilizzo delle API di Azure Digital Twin, ma in un futuro sarà semplice aggiungere nuove funzionalità grazie alle potenti API per le query che Azure mette a disposizione.

### **3.2.3 Simulazione dati di telemetria in arrivo dal GPS**

Non avendo a disposizione sensori reali si è scelto di simularli tramite una semplice app in C#. L'applicazione *DeviceSimulator* produce un flusso di telemetrie e le invia all'IoTHub, proprio come avverrebbe tramite un sensore reale. L'applicazione richiede la stringa di connessione all'IoTHub e la stringa di connessione relativa al

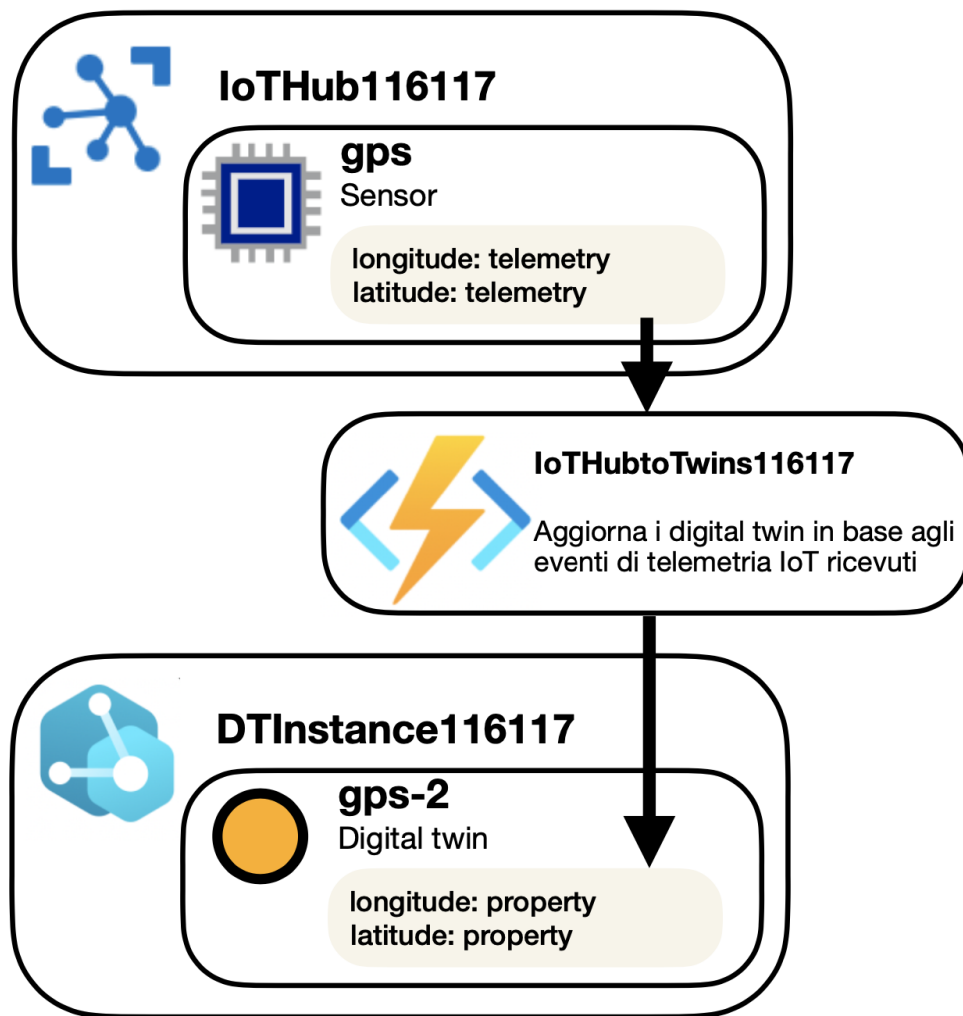


Figura 5: Ogni volta che viene inviato un evento di telemetria dal device GPS, una function app elabora le telemetrie e aggiorna le proprietà del digital twin

device presente all'interno dell'IoTHub. L'applicazione invia nuove coordinate al device presente nell'IoTHub ogni 3000 millisecondi.

Per aggiornare automaticamente i digital twin con i dati di telemetria in arrivo bisogna connettere la digital twin instance all'IoTHub. Di seguito sono riportati i passaggi svolti:

- **Creare una function app:** serve creare una funzione per accedere ai digital twins e aggiornare i device dei twins in base agli eventi di telemetria ricevuti.

- Creare la funzione tramite Visual Studio di tipo Event Grid trigger.
- Pubblicare la function app in Azure.

- **Configurare la function app:**

- Configurare l'identità gestita in modo che l'app possa accedere alla digital twin instance.

---

```
az functionapp identity assign --resource-group
    ResourceGroup116117 name IoTHubtoTwins116117
```

---

- Assegnare l'identità della function app al ruolo Azure Digital Twins Data Owner per la digital twin instance.

---

```
az dt role-assignment create --dt-name DTInstance116117
    --assignee "2d8b6f01-1993-48aa-a7cf-00b895484207"
    --role "Azure Digital Twins Data Owner"
```

---

- Rendere l'URL della digital twin instance accessibile dalla function app impostando una variabile d'ambiente per la function.

---

```
az functionapp config appsettings set --resource-group
    ResourceGroup116117 --name IoTHubtoTwins116117
    --settings ADT_SERVICE_URL=
    https://DTInstance116117.api.weu.digitaltwins.azure.net
```

---

- **Connettere la function app all'IoTHub**

- Creare un'event subscription: nell'istanza IoTHub116117, in Eventi, creare una event subscription selezionando l'end-point per scegliere la funzione di Azure da usare.

### 3.2.4 Time Series Insight

Time Series Insights è un servizio che permette di fare analisi. Grazie ad esso è possibile raccogliere, elaborare, archiviare e analizzare i dati provenienti dai digital twin. Questo strumento è molto utile nel caso di ambienti IoT che producono



miriadi di dati difficili da analizzare. Tramite TSI è possibile identificare anomalie nascoste dalla grande mole di dati, che, altrimenti, non sarebbero emerse.

Nel nostro dominio, al momento, non è previsto un grande flusso di telemetrie in entrata, per questo motivo Time Series Insights non è sfruttato in tutte le sue potenzialità.

In un'integrazione futura del sistema con un maggior numero di sensori, Time Series Insights potrebbe essere un punto chiave per l'ottimizzazione del sistema fisico, permettendo tecniche di analisi come la descriptive analytics oppure la predictive analytics.

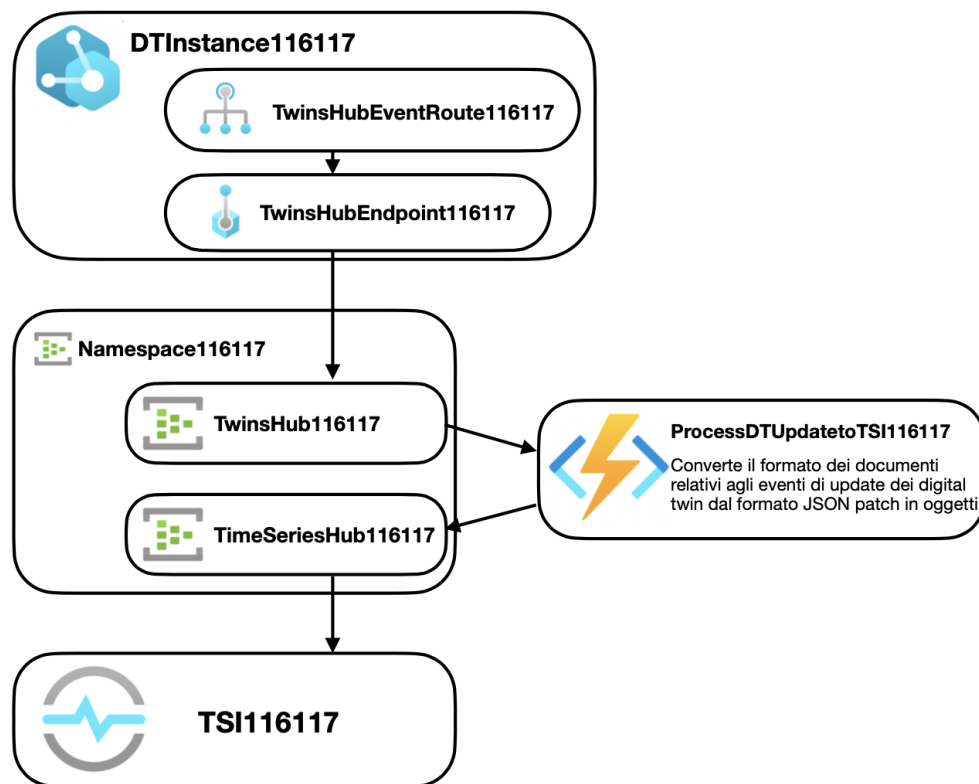


Figura 6: Collegamento della TSI instance con la digital twin instance

Di seguito è riportato come è stato costruito l'ambiente Time Series Insights all'interno del nostro sistema.

- **Namespace116117**: è utilizzato per contenere i due event hub necessari, ovvero timeserieshub116117 ed twinshub116117. Creato con il seguente

comando, specificando il resource group a cui deve appartenere.

---

```
az eventhubs namespace create --name Namespace116117
--resource-group ResourceGroup116117 --location westeurope
```

---

- **TwinsHub116117:** è l'event hub utilizzato per ricevere gli eventi di cambiamento dei digital twins. Creato con i seguenti passi:

- **Creazione dell'event hub:**

---

```
az eventhubs eventhub create --name TwinsHub116117
--resource-group ResourceGroup116117 --namespace-name
Namespace116117
```

---

- **Creazione dell'authorization rule:** bisogna creare un authorization rule con autorizzazione di invio e ricezione:

---

```
az eventhubs eventhub authorization-rule create --rights
Listen Send --name TwinsHubAuthRule116117
--resource-group ResourceGroup116117 --namespace-name
Namespace116117 --eventhub-name TwinsHub116117
```

---

- **Creazione dell'endPoint:** si crea un endpoint che collega il TwinsHub116117 alla digital twin instance:

---

```
az dt endpoint create eventhub --dt-name DTInstance116117
--eventhub-resource-group ResourceGroup116117
--eventhub-namespace Namespace116117 --eventhub
TwinsHub116117 --eventhub-policy TwinsHubAuthRule116117
--endpoint-name TwinsHubEndpoint116117
```

---

- **Creazione dell'event route:** la digital twin instance emette eventi di aggiornamento dei twin ogni volta che il loro stato cambia. L'event route instraderà questi eventi di aggiornamento al TwinsHub116117 per un'ulteriore elaborazione. L'event route invierà gli eventi di aggiornamento dei twin all'endpoint creato precedentemente. Il filtro di questa route permetterà di passare all'endpoint solo i messaggi di update.

---

```
az dt route create --dt-name DTInstance116117 --endpoint-name
TwinsHubEndpoint116117 --route-name
TwinsHubEventRoute116117 --filter "type =
'Microsoft.DigitalTwins.Twin.Update'"
```

---

- **Ottenere la stringa di connessione al TwinsHub116117:** la stringa di connessione servirà successivamente per configurare la function app

---

```
az eventhubs eventhub authorization-rule keys list
--resource-group ResourceGroup116117 --namespace-name
Namespace116117 --eventhub-name TwinsHub116117 --name
TwinsHubAuthRule116117
```

---

- **TimeSeriesHub116117:** event hub utilizzato per trasmettere gli eventi a Time Series Insights

- **Creazione dell'event hub:**

---

```
az eventhubs eventhub create --name TimeSeriesHub116117
--resource-group ResourceGroup116117 --namespace-name
Namespace116117
```

---

- **Creazione dell'Authorization rule:**

---

```
az eventhubs eventhub authorization-rule create --rights
Listen Send --name TimeSeriesHubAuthRule116117
--resource-group ResourceGroup116117 --namespace-name
Namespace116117 --eventhub-name TimeSeriesHub116117
```

---

- **Ottenere la stringa di connessione al TimeSeriesHub116117:** la stringa di connessione servirà successivamente per configurare la function app

---

```
az eventhubs eventhub authorization-rule keys list
--resource-group ResourceGroup116117 --namespace-name
```

```
Namespace116117 --eventhub-name TimeSeriesHub116117 --name  
TimeSeriesHubAuthRule116117
```

---

- **ProcessDTUpdatetoTSI116117**: function app che converte il formato dei documenti relativi agli eventi di update dei digital twins dal formato JSON patch (che contiene solo i valori aggiornati ed aggiunti ai digital twins) in oggetti JSON.

- Creare una function app di tipo event hub trigger in Visual Studio e pubblicarla in Azure.
- Assegnare i ruoli di accesso in modo che possa accedere alla digital twin instance.
- Configurare le impostazioni della function app per il TwinsHub116117 utilizzando la primaryConnectionString ottenuta in precedenza:

---

```
az functionapp config appsettings set --settings  
    "EventHubAppSetting-Twins=Endpoint=sb://namespace116117.  
servicebus.windows.net/;SharedAccessKeyName=TwinsHubAuthRule116117;  
SharedAccessKey=V/r4GocdqHmpkXRxZQGHPLab4FHJuBcId3h3FTVW2Eo=;  
EntityPath=TwinsHub116117" --resource-group  
    ResourceGroup116117 --name ProcessDTUpdatetoTSI116117
```

---

- Configurare le impostazioni della function app per il TimeSeriesHub116117 utilizzando la primaryConnectionString ottenuta in precedenza:

---

```
az functionapp config appsettings set --settings  
    "EventHubAppSetting-TSI=Endpoint=sb://namespace116117.servicebus.  
windows.net/;SharedAccessKeyName=TimeSeriesHubAuthRule116117;SharedAccessKey=  
vFLQmzQAQUHhfVV+aCg8xl70ey9mPUVe3bKun8FE7ak=;EntityPath=TimeSeriesHub116117"  
    --resource-group ResourceGroup116117 --name  
    ProcessDTUpdatetoTSI116117
```

---

- **TSI116117**: istanza di Time Series Insights. L'istanza è stata creata dall'Azure Portal. Durante la creazione viene richiesto il **nome proprietà** che corrisponde al Time Series ID e non è modificabile in seguito, nel nostro caso

è stato scelto il **dtId** ovvero l'identificatore univoco di ogni digital twin. Inoltre bisogna specificare l'event source, nel nostro caso gli eventi provengono dall'event hub creato in precedenza.

- **Visualizzazione dei dati:** per visualizzare i dati in Time Series Insights è stato usato TSIExplorer. Esso consente di analizzare, archiviare e visualizzare tutti gli eventi in arrivo dall'hub.

### 3.3 Come avviene lo shadowing

I dati relativi ai digital twins vengono inseriti interamente a mano, attraverso l'intervento degli operatori che settano le varie informazioni di cui interessa tenere traccia.

L'unica eccezione è data dai dati provenienti dal GPS ovvero, latitudine e longitudine. Questi dati vengono aggiornati come descritto nel sottocapitolo 3.2.3.

### 3.4 Digital Twin API

Azure mette a disposizione un ampio set di API, queste 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.

Entrambi i tipi di API possono essere utilizzate in svariati modi, come ad esempio:

- (C) .NET SDK
- Java SDK
- JavaScript SDK
- Python SDK
- Go SDK

- Da riga di comando
- Dall’Azure portal

Nel caso specifico del progetto Trasporti116117, le **Control Plane API** sono state utilizzate da riga di comando e dall’Azure portal per creare e configurare tutte le risorse necessarie, riportate di seguito:

- **ResourceGroup116117**: Resource Group che contiene tutte le seguenti risorse.
- **DTInstance116117**: digital twin instance.
- **IoTHub116117**: IoT Hub.
- **Namespace116117**: Namespace che contiene i due event hub TimeSeriesHub116117 e TwinHub116117.
- **IoTHubtoTwins116117**: Function App che permette di inserire i dati di telemetria dell’hub IoT nella digital twin instance.
- **ProcessDTUpdateToTSI116117**: Function App per il funzionamento di Time Series Insights.
- **TSI116117**: Istanza di Time Series Insights.
- **EventSource116117**: Event source per Time Series Insights.
- **EventHubConsumerGroup116117**: meccanismo di pubblicazione/sottoscrizione di Hub eventi utilizzato per Time Series Insights.

Per quanto riguarda le **Data Plane API** sono state utilizzare principalmente tramite il Java SDK per la costruzione del CallCenter Client e del AmbulanceClient come riportato nella sezione seguente.

### 3.5 Implementazione dei servizi forniti dai Digital Twins

Tramite le API fornite da Azure sono stati implementati diversi servizi utili ai client sviluppati. Di seguito riportate le classi principali che li implementano:

- **Client**: classe utilizzata da tutti gli altri servizi, crea la connessione al client e fornisce lo stesso ogni volta che viene richiesto.

Per l'autenticazione, Azure mette a disposizione Azure Identity. *ClientSecretCredentialBuilder* permette l'autenticazione ad Azure Active Directory tramite un ClientSecret generato per un'applicazione client creata in precedenza da AAD. Oltre al ClientSecret bisogna fornire il tenantId (reperibile da AAD) ed il clientId corrispondente all'id dell'applicazione creata in AAD. Oltre alle credenziali per l'autenticazione è necessario inserire un endpoint, nel nostro caso è *https://DTInstance116117.api.weu.digitaltwins.azure.net* ovvero il nome dell'host della digital twin instance.

---

```
client = new DigitalTwinsClientBuilder()
    .credential(
        new ClientSecretCredentialBuilder()
            .tenantId(Constants.TENANT_ID)
            .clientId(Constants.CLIENT_ID)
            .clientSecret(Constants.CLIENT_SECRET)
            .build()
    )
    .endpoint(Constants.ENDPOINT)
    .buildClient();
```

---

- **Create service**: Sono stati implementati una serie di servizi che permettono la creazione di nuovi digital twins:
  - **createAmbulance**
  - **createBookingTransport**
  - **createOperator**
  - **createPatient**
  - **createInfoRequest**
  - **createServiceRequest**
  - **startTransport**

Nello snippet di codice seguente viene creato un digital twin di tipo *BasicDigitalTwin* che richiede come parametro un id univoco. Successivamente tramite *setMetadata* si setta il modello che questo digital twin dovrà avere e con *addToContents* si possono aggiungere ulteriori proprietà al digital twin. Questo è un JSON object che descrive il digital twin che si vuole creare.

Per creare effettivamente il digital twin bisogna utilizzare *createOrReplaceDigitalTwin*, fornito dal client. Esso richiede in input un id univoco, il JSON object che rappresenta il digital twin da creare (ovvero *ambulanceDT*) ed infine il tipo della classe per serializzare la richiesta. Nel caso in cui venisse inserito un id già presente per un altro digital twin quest'ultimo verrà sostituito dal nuovo.

---

```
BasicDigitalTwin ambulanceDT = new
    BasicDigitalTwin(ambulanceId.getAmbulanceId())
    .setMetadata(
        new BasicDigitalTwinMetadata()
        .setModelId(Constants.AMBULANCE_MODEL_ID))
    .addToContents("number", ambulanceNumber);

Client.getClient().createOrReplaceDigitalTwin(ambulanceId.getAmbulanceId(),
    ambulanceDT, BasicDigitalTwin.class);
```

---

Successivamente viene creata una relazione tra due twins, con la stessa logica con cui viene creato il digital twin nello snippet precedente. *createOrReplaceRelationship* richiede in input l'id del source digital twin e l'id della relazione. Inoltre, richiede un oggetto di tipo *BasicRelationship* ovvero il JSON object che descrive la relazione. Esso dovrà specificare l'id della relazione, il digital twin di partenza della relazione e quello di arrivo ed infine il nome della relazione, indicato nei modelli.

---

```
BasicRelationship createdRelationship =
Client.getClient().createOrReplaceRelationship(
    ambulanceId.getAmbulanceId(),
    ambulanceId.getAmbulanceId() + "to" + ambulanceId.getGpsId(),
    new BasicRelationship(
```



```
        ambulanceId.getAmbulanceId() + "to" +  
            ambulanceId.getGpsId(),  
        ambulanceId.getAmbulanceId(),  
        ambulanceId.getGpsId(),  
        "contains"),  
        BasicRelationship.class);
```

---

- **Delete service:** sono stati implementati una serie di servizi che permettono la cancellazione di alcuni digital twins:

- **deleteAmbulance**
- **deleteBookingTransport**
- **deleteOperator**
- **deletePatient**
- **deleteServiceRequest**
- **deleteInfoRequest**
- **deleteTransport**

Solamente i *deleteAmbulance* e *deleteBookingTransport* sono effettivamente utilizzati all'interno del client, poiché si è ritenuto più opportuno mantenere i restanti digital twins a medio lungo termine per permettere l'analisi. La cancellazione di un digital twin avviene sempre mediante il client, nello snippet di codice seguente per prima cosa si cancellano tutte le relazioni che hanno come source il digital twin che deve essere cancellato e successivamente si elimina il digital twin. In questo caso specifico viene eliminato sia il twin relativo all'ambulanza sia quello relativo al GPS poiché non può esistere un GPS senza un ambulanza. Questo snippet di codice restituisce uno status code *DELETED* in caso di effettiva cancellazione del twin. Può, però, capitare che il digital twin che si sta tentando di cancellare abbia relazioni attive dove esso è il target, in questo caso viene restituito uno status code *RELATION\_EXISTING*, che indica che il twin non è stato eliminato poiché ha relazioni attive.

---

```
try{
```

```

        Client.getClient().listRelationships(ambulanceId.getAmbulanceId(),
            BasicRelationship.class)
            .forEach(rel ->
                Client.getClient().deleteRelationship(ambulanceId.getAmbulanceId(),
                    rel.getId()));
        Client.getClient().deleteDigitalTwin(ambulanceId.getAmbulanceId());
        Client.getClient().deleteDigitalTwin(ambulanceId.getGpsId());

        return DeleteDigitalTwinStatusCode.DELETED;
    } catch (ErrorResponseException e){
        if(e.getLocalizedMessage().contains(
            AzureErrorMessage.RELATIONSHIP_NOT_DELETED)) {
            return DeleteDigitalTwinStatusCode.RELATION_EXISTING;
        }
        throw e;
    }
}

```

---

- **Query service:** sono stati implementati una serie di servizi che permettono di interrogare i digital twins. Azure mette a disposizione un linguaggio di query molto potente che permette di interrogare il digital twin graph. Di seguito vengono riportati i servizi implementati per i due client costruiti:
  - **getAllAmbulanceIdTwins:** restituisce tutte le ambulanze nel digital twin graph.
  - **getAllBookingId:** restituisce tutte le prenotazioni nel digital twin graph.
  - **getAllBookingToDoForTheDay:** restituisce tutte le prenotazioni non ancora prese in carico per una data giornata.
  - **getAllBookingForTheDay:** restituisce tutte le prenotazioni per una data giornata, sia quelle già svolte che quelle non ancora prese in carico.
  - **getAllBookingForToday:** restituisce tutte le prenotazioni per il giorno corrente, sia quelle già svolte che quelle non ancora prese in carico.
  - **getGPSCoordinatesOfAmbulance:** restituisce la posizione, in coordinate, di una data ambulanza.

- **getAllOperatorId**: restituisce tutti gli operatori nel digital twin graph.
- **getPatientIdByBookingId**: restituisce il paziente associato ad una data prenotazione.
- **getAllPatientId**: restituisce tutti i pazienti nel digital twin graph.
- **getAllServiceRequestId**: restituisce tutte le richieste di servizio nel digital twin graph.
- **getAllInfoRequestId**: restituisce tutte le richieste di informazioni nel digital twin graph.
- **getRouteByBookingId**: restituisce l'itinerario associato ad una data prenotazione.
- **getAllTransportId**: restituisce tutti i trasporti nel digital twin graph.
- **getTransportOfAmbulance**: restituisce tutti i trasporti relativi ad una data ambulanza.
- **getAllTransportInProgress**: restituisce tutti i trasporti in corso al momento.
- **setTakeOwnership**: setta la presa in carico di una prenotazione, ovvero l'inizio del trasporto relativo a quella prenotazione.
- **transportEnded**: setta la fine di un trasporto.

Di seguito è riportato un esempio di query. Questa query restituisce tutti i trasporti in corso. Il client fornisce un metodo *query()* che prende come argomento la query e il modello della classe in cui deserializzare la risposta della query. Restituisce una *PagedIterable* che permette di iterare sulle *PagedResponse* che contengono i risultati della query.

---

```

ArrayList<TransportId> transportIds = new ArrayList<>();
String query = "SELECT * " +
               "FROM DIGITALTWINS " +
               "WHERE IS_OF_MODEL(' " + Constants.TRANSPORT_MODEL_ID
               + "') AND NOT IS_DEFINED ( endDateTime )";

Client.getClient().query(query, BasicDigitalTwin.class)

```

```
        .forEach(r-> transportIds.add(new TransportId(r.getId())));  
return transportIds;
```

---

## 4 Validazione e Test

### 4.1 Testing automatizzato

Lo sviluppo dei servizi forniti ai client è stato testato mediante la libreria JUnit. Il progetto è stato testato su sistemi Linux e Mac in modo automatico mediante l'utilizzo di Travis CI.

All'interno del progetto sono presenti le seguenti classi di test:

- `AmbulanceIdGeneration`: testa la generazione degli id di ambulanza e GPS.
- `DtAmbulance`: testa i servizi relativi al digital twin dell'ambulanza.
- `DtBookingTransport`: testa i servizi relativi al digital twin della prenotazione.
- `DtConnection`: testa la connessione al client.
- `DtInfoRequest`: testa i servizi relativi al digital twin della richiesta di informazioni.
- `DtOperator`: testa i servizi relativi al digital twin dell'operatore.
- `DtPatient`: testa i servizi relativi al digital twin del paziente.
- `DtServiceRequest`: testa i servizi relativi al digital twin della richiesta di servizio.
- `DtTransport`: testa i servizi relativi al digital twin del trasporto.

### 4.2 Implementazione Scenari

#### 4.2.1 Scenario 1

L'operatore che riceve la chiamata inserisce nel sistema il nuovo paziente e la prenotazione, tramite il Call Center Client. Nell'immagine seguente si può vedere l'inserimento della prenotazione e del paziente, poiché non era ancora all'interno del sistema.

Aggiungi Paziente

Nome

Mario

Cognome

Rossi

Data di nascita

02/08/1945

Via

Via Bologna

Numero

16

Città

Gambettola

Provincia

FC

Cap

47035

Codice fiscale

MRARSS45H17G187U

Stato di salute

Il paziente può stare solo in posizior

Autonomia

NOT\_AUTONOMOUS

Annulla

OK

Richiesta di servizio

Partenza

Via

Via Giovanni Ghirotto

Numero

286

Città

Cesena

Provincia

FC

Cap

47521

Arrivo

Via

Via Bologna

Numero

16

Città

Gambettola

Provincia

FC

Cap

47035

Data trasporto

16/06/2021

Ora trasporto

16.00

Paziente

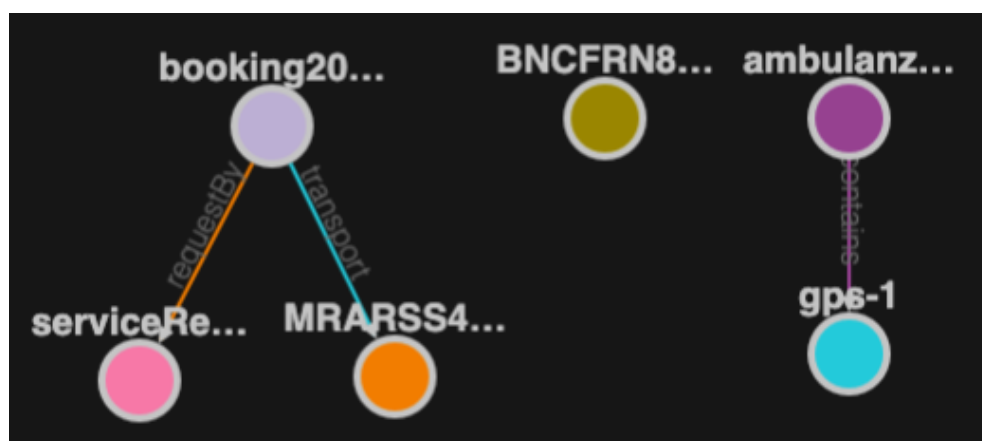
MRARSS45H17G187U

Aggiungi prenotazione

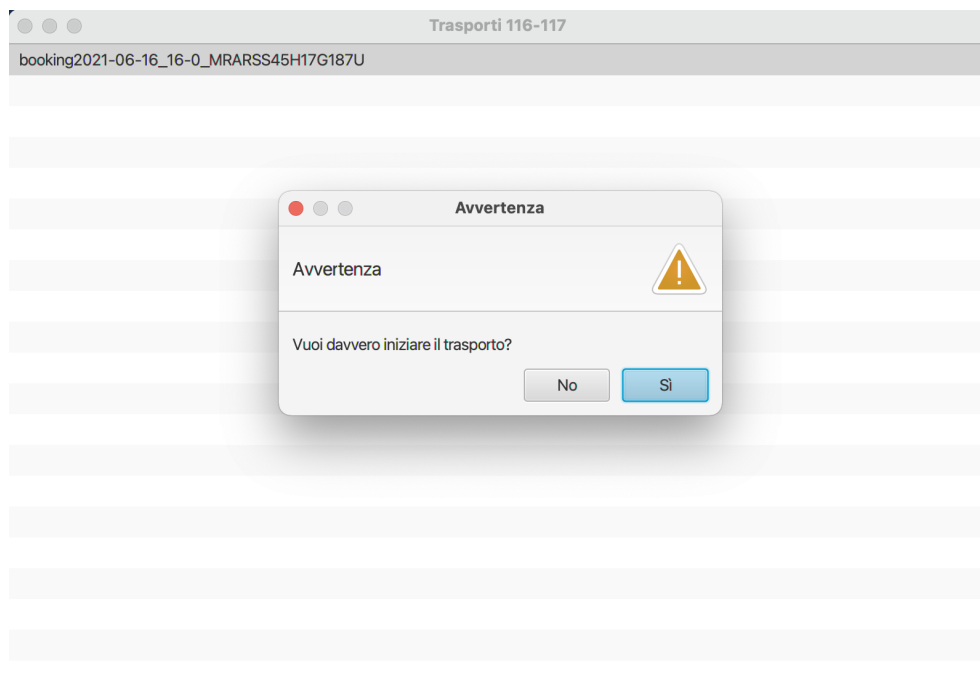
Registra richiesta di servizio senza prenotazione

Annulla

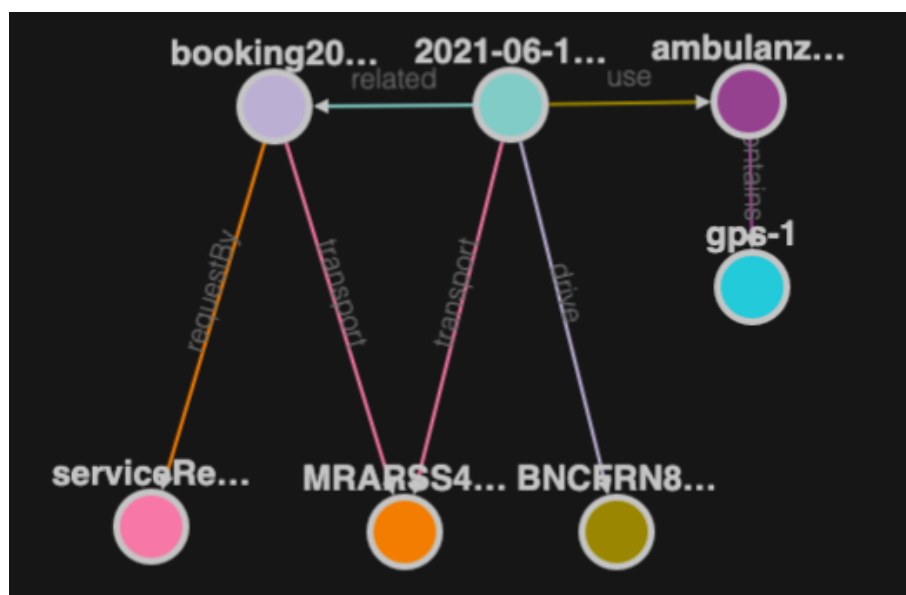
Nella figura seguente è riportato il digital twin graph dopo l’inserimento di questi dati:



L’operatore controlla le prenotazioni per la giornata dall’Ambulance client. Successivamente, seleziona la prenotazione e conferma la presa in carico del trasporto come nella figura seguente:



A questo punto esegue il trasporto. Giunto a destinazione, dopo avere lasciato il paziente a destinazione, termina il trasporto. In figura si può vedere il digital twin graph al termine dello scenario 1.



#### **4.2.2 Altri scenari**

La simulazione degli altri 3 scenari è risultata pressoché uguale allo scenario 1. Durante le simulazioni non sono emersi problemi rilevanti.

Durante lo svolgimento delle simulazioni è sorta, però, una possibile ottimizzazione da svolgere negli sviluppi futuri: le prenotazioni potrebbero essere proposte all'operatore dell'ambulanza in ordine di vicinanza al mezzo che sta utilizzando, in modo da ottimizzare lo sfruttamento dei mezzi e degli operatori.



## 5 Discussione e conclusione

### 5.1 Conoscenze acquisite da questa sperimentazione

Durante questo progetto abbiamo avuto la possibilità di sperimentare l'utilizzo dell'approccio basato su digital twin. Riteniamo che questo approccio può rendere il massimo nello studio di sistemi complessi, in particolare in sistemi che ricevono un grande flusso di dati, spesso non contestualizzati come, ad esempio, succede nell'IIoT. In questo caso, i digital twins diventano un potente strumento di analisi, in quanto permettono di strutturare e contestualizzare i flussi di dati ricevuti. Inoltre, tramite essi, è possibile svolgere predizioni, utili a ottimizzare il loro asset fisico.

#### 5.1.1 Azure Digital Twins

Durante questa sperimentazione, per l'utilizzo dei digital twins, ci siamo appoggiati alla piattaforma di Azure Digital Twin.

Abbiamo esplorato il mondo di Azure, provando un'esperienza di sviluppo in cloud, attività non approfondita a livello pratico fino ad ora.

Per lo sviluppo di un ecosistema di digital twin, non è stato utilizzato solo il servizio Azure Digital Twin, ma abbiamo avuto la possibilità di utilizzare altri servizi offerti da Azure come ad esempio Azure IoT, che contiene una raccolta di servizi per connettere e gestire asset IoT, Azure Identity e Azure Active Directory che è un servizio di gestione delle identità e degli accessi.

Azure Digital Twin nel complesso è uno strumento completo, scalabile e molto potente. A nostro parere è molto utile in grandi realtà, dove i flussi di dati sono considerevoli e dove analisi e predizioni sono particolarmente complicate. L'utilizzo in piccole organizzazioni, dal nostro punto di vista, è sconsigliabile in quanto ha un costo elevato e in una piccola realtà non si sfrutterebbero a pieno le sue funzionalità.

In definitiva, ci riteniamo molto soddisfatte dalle conoscenze apprese da questo progetto che ci ha permesso di lavorare in cloud, esplorare la piattaforma di Azure ed approfondire e sperimentare l'utilizzo dei digital twins.

## **5.2 Efficacia dell’approccio basato su Digital Twin applicato al dominio Trasporti 116117**

Per quanto riguarda il dominio relativo ai trasporti secondari 116117 questo approccio può essere, potenzialmente, un punto di svolta per la risoluzione dei problemi sollevati durante gli incontri con l’esperto di dominio.

Tuttavia, riteniamo che, data l’analisi di dominio fatta in parte su dati che rispecchiano il dominio reale e in parte su dati ipotizzati da noi, ne è risultata un’analisi, a nostro parere, semplificata rispetto alla situazione reale.

Considerato questo fattore, il progetto risultante non ha esaltato pienamente i benefici che in realtà si possono ottenere utilizzando un approccio potente come i digital twins.

## **5.3 Aspetti rilevanti non affrontati e miglioramenti futuri**

### **5.3.1 Gestione delle ambulanze**

Un aspetto piuttosto importante da considerare in questo sistema, è quello della segnalazione, da parte dello stesso, della presenza o meno di un’ambulanza disponibile per la data indicata, già al momento della richiesta del servizio.

Al momento in cui un utente richiede di poter usufruire di un servizio in una determinata data, potrebbe essere utile che il sistema suggerisse automaticamente all’operatore se in quel determinato momento c’è un mezzo ancora disponibile o se, viceversa, sono già tutti prenotati. In questo modo verrebbe automatizzato maggiormente il compito dell’operatore che risponde alla chiamata.

### **5.3.2 Ottimizzazione dei percorsi**

Durante la fase di valutazione è emersa una possibile ottimizzazione relativa ai percorsi svolti dalle ambulanze. In un’ottica di gestione unificata delle varie sedi del 116117 sarebbe auspicabile memorizzare un’ulteriore informazione relativa all’ambulanza, che riguarda la sua sede di appartenenza.

Questo, insieme, al servizio di tracciamento dell’ambulanza già presente, potrebbe essere utilizzato per ottimizzare l’impiego dei mezzi a disposizione. Ovvero, le prenotazioni potrebbero essere proposte ad hoc all’operatore in base alla posizione attuale ed alla sede di appartenenza.

Si potrebbe applicare un algoritmo di ricerca operativa che permetta, date le prenotazioni dei trasporti relativi ad una giornata e i luoghi di appartenenza dell'ambulanza stessa, di assegnare ad ogni ambulanza un opportuno insieme di trasporti da eseguire in modo da ottimizzarne i percorsi e ridurre gli sprechi di risorse.

### **5.3.3 Analisi, simulazioni e predizioni**

Allo stato attuale, il sistema permette di analizzare i dati tramite Time Series Insights. Questo strumento permette di migliorare i processi decisionali all'interno dell'organizzazione, interrogando e contestualizzando dati tramite gerarchie, relazioni e proprietà. Azure mette a disposizione, inoltre, strumenti di predictive analytics come ad esempio Azure Machine Learning che mette a disposizione una serie di moduli per creare un modello predittivo. Questo argomento non è stato però approfondito durante lo svolgimento del progetto.

### **5.3.4 Archiviazione dei dati**

Un'alternativa all'attuale gestione dell'archiviazione dei dati potrebbe essere quella di utilizzare un database esterno.

Attualmente, tutti i dati vengono mantenuti memorizzati all'interno dei digital twins, sia che siano attuali, sia che facciano parte dello storico delle informazioni. Si potrebbe, quindi, pensare di archiviare all'interno di un database esterno i dati storici relativi ai servizi già conclusi da un predeterminato lasso di tempo e che si considerano non più utili per svolgere le analisi sul servizio.