



ATTICUS

Ambient-intelligent Tele-monitoring System

Codice progetto: ARS01_00860

Deliverable D6.3

Documento dell'architettura del sistema

(System Design Document)



**UNIVERSITÀ
DEGLI STUDI
DEL MOLISE**



Consorzio Italia



Data: Marzo 2020

| | |
|--|--|
| Leader OR | DISC |
| Obiettivo realizzativo (OR) | OR3 – Analisi dei requisiti del sistema ATTICUS |
| Attività | 6.3 – Progettazione dell'architettura del sistema |
| Descrizione dell'attività <i>(come da capitolato)</i> | Questa attività prevede la progettazione del sistema e, quindi, la definizione della sua architettura. Si dividerà il sistema in sottosistemi e si definiranno precisamente le interfacce tra questi, e la comunicazione con le altre componenti di ATTICUS. L'architettura del sistema sarà disegnata seguendo i principi del "Secure by design and by default" in modo da soddisfare l'Art. 25 del nuovo regolamento EU 2016/679. A tal proposito sia lo standard ISO 27001 sia il relativo sub-standard ISO 27034 saranno utilizzati come guida pratica per un efficiente design sicuro del sistema di supporto alle decisioni. |
| Partner coinvolti nell'attività | <ul style="list-style-type: none"> ● UNIMOL ● UNISANNIO |
| Stato del documento | Final draft |
| Confidenzialità | Ristretta |

Indice del documento

| | | |
|-------|---|-----|
| 1 | Introduzione | 7 |
| 2 | Architettura del sistema | 9 |
| 2.1 | Architettura logica | 9 |
| 2.1.1 | Tecnologie..... | 11 |
| 2.1.2 | Presentation layer..... | 14 |
| 2.1.3 | Application layer | 15 |
| 2.1.4 | Database layer | 18 |
| 2.2 | Architettura fisica | 20 |
| 2.2.1 | Strategia di deployment | 21 |
| 2.2.2 | Diagramma di deployment | 25 |
| 2.2.3 | Configurazione di deployment..... | 26 |
| 2.2.4 | Continuous machine learning | 28 |
| 3 | Servizi del software di monitoraggio | 32 |
| 3.1 | Servizi gestione utenti | 32 |
| 3.2 | Servizi gestione kit di monitoraggio | 71 |
| 3.3 | Servizi arruolamento | 95 |
| 3.4 | Servizi gestione assistiti | 103 |
| 3.5 | Servizi gestione monitoraggio on demand | 156 |
| 3.6 | Servizi gestione allarmi..... | 175 |
| 4 | Servizi del DSS di livello 3 | 193 |
| 4.1 | Servizi dei moduli esterni al DSS..... | 194 |
| 5 | Struttura del DSS di livello 2 | 198 |
| 6 | Struttura dell'App MyATTICUS | 199 |

| | | |
|-------|---|-----|
| 6.1 | Architettura dell'App MyATTICUS | 199 |
| 6.2 | Rilevazione delle anomalie nell'App MyATTICUS | 200 |
| 7 | Sicurezza del sistema | 203 |
| 7.1 | Architettura secure-by-design e by-default..... | 203 |
| 7.1.1 | Conformità al GDPR (UE 2016/679) | 203 |
| 7.1.2 | Standard di riferimento (ISO 27001 e 27034) | 205 |
| 7.2 | Autenticazione dei servizi..... | 206 |
| 7.2.1 | Role Based Access Control (RBAC) | 206 |
| 7.2.2 | JSON Web Token (JWT)..... | 207 |
| 7.3 | Sicurezza della comunicazione | 208 |
| 7.4 | Sicurezza dei dati | 209 |
| | Riferimenti bibliografici..... | 211 |

Indice delle tabelle

| | |
|---|----|
| Tabella 1. Vincoli dell'approccio architetturale REST..... | 21 |
| Tabella 2. Versione del software utilizzato per il deploy | 28 |

Indice delle figure

| | |
|---|-----|
| Figura 1. Architettura a tre livelli (Three Tier Architecture)..... | 10 |
| Figura 2. Moduli del DSS di livello 3. | 12 |
| Figura 3. Tecnologie e framework del sistema ATTICUS. | 12 |
| Figura 4. Presentation layer del software di monitoraggio in ATTICUS. | 14 |
| Figura 5. Application layer di software di monitoraggio e DSS di livello 3. | 16 |
| Figura 6. Database layer del sistema ATTICUS | 19 |
| Figura 7. Azioni eseguibili sulle risorse dal sistema ATTICUS. | 23 |
| Figura 8. Evoluzione nella gestione delle infrastrutture informatiche (fonte: kubernetes.io)..... | 24 |
| Figura 9. Diagramma di deployment del software di monitoraggio e del DSS di livello 3. | 25 |
| Figura 10. Rappresentazione del processo di deployment tramite pipeline..... | 27 |
| Figura 11. Schema dei repository usati dai detector basati su machine learning..... | 29 |
| Figura 12. Schema del processo di aggiornamento dei modelli predittivi. | 31 |
| Figura 13. Struttura del DSS di livello 2 | 198 |
| Figura 14. Struttura dell'App MyATTICUS | 200 |
| Figura 15. Schermata di Login (sinistra) e schermata home (destra) dell'App MyATTICUS | 201 |
| Figura 16. Impostazione della posizione dell'abitazione su MyATTICUS..... | 202 |

1 Introduzione

Questo documento presenta l'architettura di due parti essenziali del sistema ATTICUS: il sistema di monitoraggio e il DSS di livello 3. Tali sistemi software, benché siano progettati in maniera modulare, così che possano essere rese facilmente indipendenti l'uno dall'altro, sono entrambi parte integrante della centrale di monitoraggio e, in ATTICUS, interagiscono tra di loro al fine di fornire le funzionalità previste dal sistema.

Il sistema di monitoraggio permette di gestire tutti gli assistiti monitorati in ATTICUS ed è la componente della centrale di monitoraggio che consente agli utenti di interagire con i dati dei pazienti, di effettuare sessioni di monitoraggio e di reagire agli allarmi generati dai sistemi di supporto alle decisioni (DSS). Il software di monitoraggio è costituito da un back-end, che si occupa di gestire le interazioni con le altre componenti del sistema (DSS di livello1, DSS di livello 2 e DSS di livello3) e con la base dati, e da un front-end, che permette, attraverso delle interfacce, di accedere a specifiche funzionalità mediche accessibili ai medici specialisti e ai responsabili di monitoraggio, mentre darà la possibilità ai caregiver e agli assistenti domiciliari di poter accedere alle informazioni di carattere medico dell'assistito e all'amministratore di sistema darà la possibilità di poter gestire gli utenti e i kit di monitoraggio.

Il DSS di livello 3 è un sistema software su cui si appoggia il software di monitoraggio per analizzare automaticamente i dati biomedici e individuare situazioni anomale relative alla salute degli assistiti. Il DSS di livello 3 è costituito da un insieme di *detector*, ognuno dei quali ha la responsabilità di individuare una specifica patologia o anomalia.

Il resto del documento è strutturato come segue. Nel capitolo 2 verrà descritta l'architettura delle due componenti; nei capitoli 3 e 4, rispettivamente, verrà presentata la divisione in sottosistemi e

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

la progettazione dei servizi del software di monitoraggio e del DSS di livello 3; nel capitolo 5 si presenteranno le tecniche adottate per garantire la sicurezza in ATTICUS.

2 Architettura del sistema

In questa sezione è presentata l'analisi del sistema da un punto di vista logico e da un punto di vista fisico. Nella parte dell'architettura logica sono analizzate a livello strutturale le relazioni tra le principali componenti software, mentre nella sezione dell'architettura fisica si specifica su che dispositivo fisico queste componenti risiedono e sono eseguite. La centrale di monitoraggio conterrà due sottosistemi: il software di monitoraggio e il DSS di livello 3. Questi due sistemi hanno responsabilità diverse: il software di monitoraggio implementa un sistema gestionale attraverso cui particolari utenti possono interagire con i dati, mentre il DSS di livello 3 rappresenta il cuore dell'intelligenza artificiale del sistema ATTICUS e ha il compito di analizzare i dati, gli avvisi di anomalia e notificare gli attori competenti attraverso il software di monitoraggio. Di seguito verranno presentate l'architettura logica e l'architettura fisica del sistema del sistema, inteso come combinazione di software di monitoraggio e DSS di livello 3.

2.1 Architettura logica

L'architettura del software di monitoraggio è di tipo *three-tier*: le funzioni sono logicamente separate nelle componenti di presentazione (*presentation layer*), di applicazione (*application layer*) e di accesso ai dati (*database layer*). L'organizzazione ad alto livello dei *layer* coinvolti in questo tipo di analisi è presentata nella Figura 1.

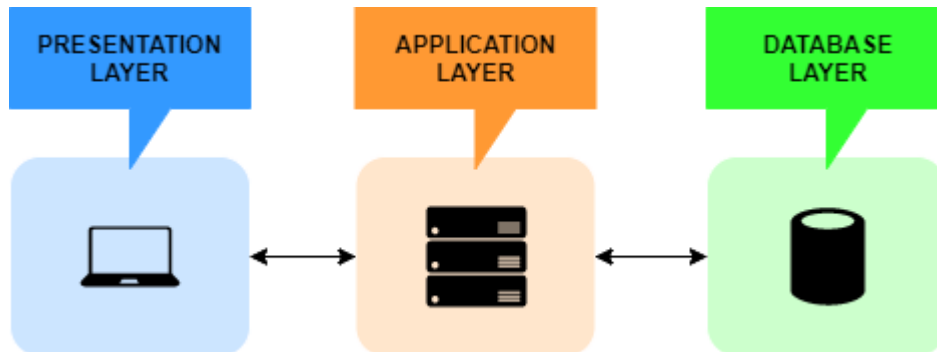


Figura 1. Architettura a tre livelli (Three Tier Architecture).

Il primo layer, il *presentation layer*, è il livello riguardante l'interfaccia grafica, la gestione degli input e degli output e la loro visualizzazione. Il sistema ATTICUS sarà disponibile su internet: gli utenti, quindi, possono accedere al sistema utilizzando un browser web. Questo componente contiene elementi HTML e mostra le informazioni ricevute dall'*application layer*. L'utente finale del sistema, con il proprio browser, si connette al web server attraverso il protocollo HTTPS.

Il livello centrale è l'*application layer*, che funge da ponte tra l'interfaccia utente e il database sottostante nascondendo i dettagli tecnici all'utente che utilizza il sistema. I componenti presenti in questo livello ricevono le richieste dal *presentation layer* e in base a queste eseguono delle operazioni di lettura o scrittura sul database.

L'ultimo livello, il *database layer*, ha come obiettivo la memorizzazione delle informazioni necessarie per il funzionamento del sistema e l'ottimizzazione dell'accesso ai dati. I dati che servono all'*application layer* sono recuperati dal database, così come i dati prodotti dall'*application layer* vengono memorizzati nel database.

Tutte le componenti di intelligenza artificiale (IA) sono implementate nel DSS di livello 3, il quale, sfruttando diversi *detector*, è in grado di confermare o annullare un avviso di anomalia proveniente dai DSS di livello inferiore. Il DSS di livello 3 è costituito da due soli layer: *application layer* e *database*

layer. Non è presente un layer relativo alla logica di presentazione, poiché questa è fornita dal software di monitoraggio. È bene notare che il DSS di livello 3 è costituito, a sua volta, da diversi moduli, ognuno dei quali implementa un singolo *detector*, ovvero una componente in grado di individuare una specifica patologia o situazione di anomalia. I *detector* sono implementati come moduli interni del DSS di livello 3, nel caso in cui non richiedano l'aggiornamento continuo di un modello di *machine learning* basato sui dati di monitoraggio degli assistiti, oppure come moduli esterni, nel caso in cui, invece, siano continuamente perfezionati in base a tali dati. In Figura 2 sono mostrati i moduli interni/esterni del DSS di livello 3.

Nei successivi paragrafi sono affrontati i tre livelli individuati con particolare riferimento alle tecnologie e framework utilizzati e il dettaglio a livello strutturale di ciascun layer.

2.1.1 Tecnologie

I tre livelli individuati nell'architettura sono una specializzazione di un tipico modello MVC (*Model-View-Controller*) e le tecnologie e i framework utilizzati sono illustrati nella Figura 3. Il Model-View-Controller è un pattern architetturale molto diffuso nello sviluppo di applicazioni web e consente di separare la logica di presentazione dalla logica di business. Di seguito i dettagli delle scelte tecnologiche.

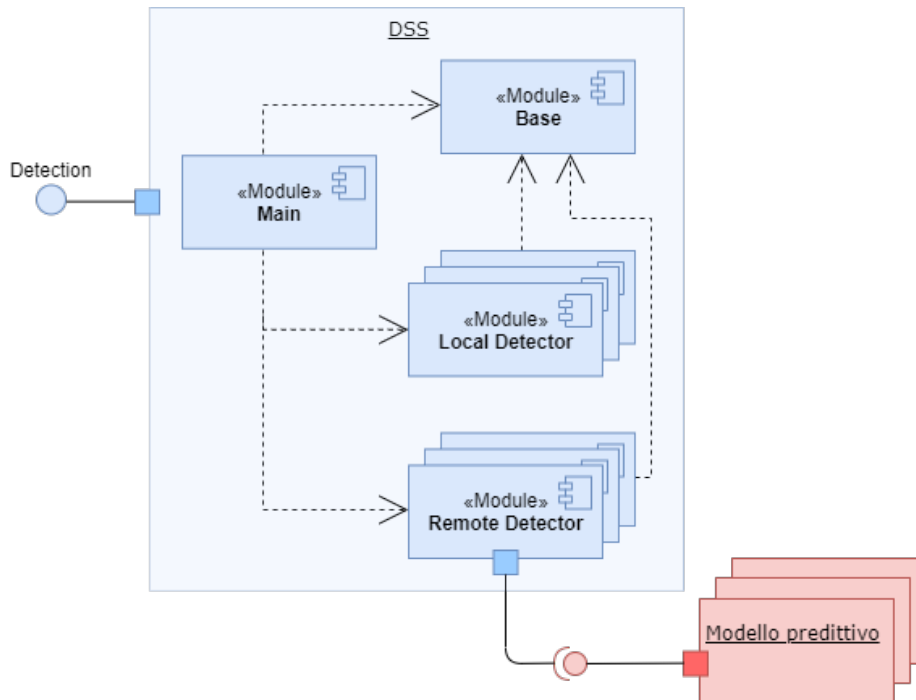


Figura 2. Moduli del DSS di livello 3.

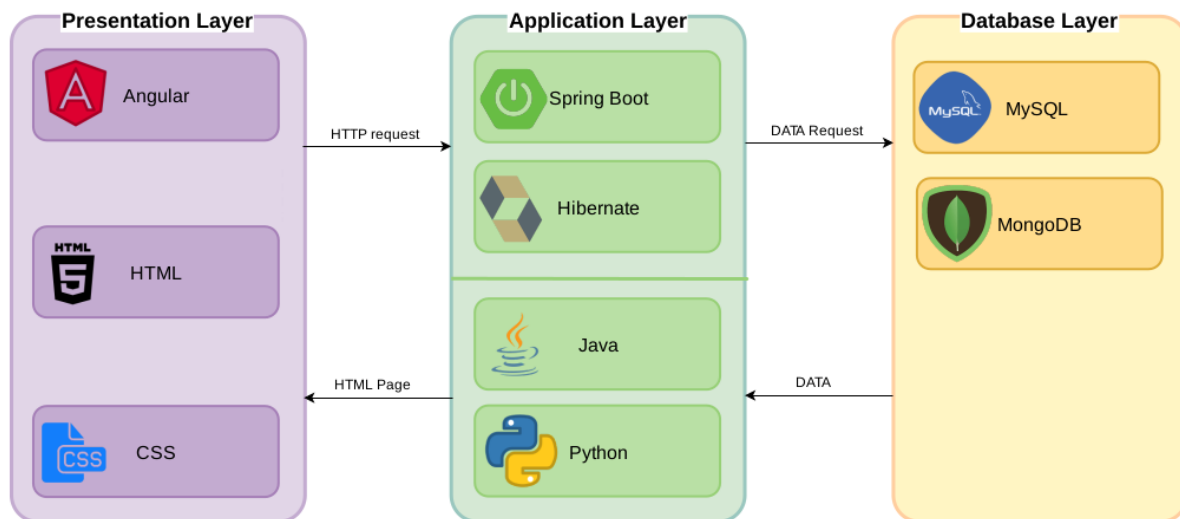


Figura 3. Tecnologie e framework del sistema ATTICUS.

Il presentation layer (*View*) è scritto in TypeScript, **HTML** e **CSS** usando il Framework **Angular**. Il layer ha la responsabilità di mostrare le informazioni e permettere l'interazione con l'utente finale della web-app in base al ruolo che ricopre. Angular è uno tra i più popolari framework open-source per sviluppare web-app. Le componenti Angular sono scritte in TypeScript, super set del linguaggio JavaScript. In particolare, TypeScript come super set estende la sintassi di JavaScript, pertanto un programma in JavaScript è anche un programma TypeScript valido quindi compatibile. Angular permette di creare *Single Page Application (SPA)*, applicazione mostrata in un web browser che non necessita di effettuare richieste a nuove pagine in modo da velocizzare l'intero processo di visualizzazione. Angular è basato su un'architettura modulare secondo la quale i componenti possono essere facilmente sostituiti, disaccoppiati o riutilizzabili in altre parti dell'applicazione.

La logica di business (*Controller*), l'*application layer*, è scritta in **Java** e rappresenta un ponte tra il layer di presentazione e il layer del database. L'*application layer* accetta richieste HTTPS dell'utente e fornisce pagine HTML sul dispositivo utilizzato. Il livello applicativo è costruito sul Framework Spring e, nello specifico, un insieme di componenti che si integra facilmente con tutto l'ecosistema, ovvero **Spring Boot**, utilizzato per la costruzione dell'applicazione standalone per la gestione degli endpoint *Representational State Transfer (REST)* illustrati nelle sezioni successive. Le richieste che arrivano dal *presentation layer* sono intercettate da specifici *handler* che prendono in carico la richiesta applicando la logica di business per l'azione ricevuta, in base all'*endpoint* chiamato. Il framework **Hibernate**, integrato in Spring, è utilizzato per gestire la persistenza e mappare oggetti Java nei database. I moduli interni del DSS di livello 3 sono implementati come moduli Java separati dal cuore del DSS stesso, in modo da renderli facilmente attivabili e disattivabili, se necessario, ed evolvibili indipendentemente da esso. I moduli esterni, invece, sono implementati in **Python** e sono richiamati dal DSS di livello 3 attraverso endpoint REST.

Il *database layer* (Model) rappresenta il sistema di archiviazione e ha il compito di memorizzare i dati. Tale layer è composto da database di tipo relazionali (**MySQL**) e database non relazionali (**MongoDB**) in base al tipo di dato che si sta manipolando. Si rimanda a D6.1 per i dettagli.

2.1.2 Presentation layer

Il *presentation layer* ha come obiettivo principale quello di interagire con l'utente e mostrare le informazioni. La Figura 4 mostra l'organizzazione architetturale delle componenti ad alto livello del *presentation layer* del software di monitoraggio di ATTICUS o, più in generale, di una tipica architettura web-app implementata in Angular.

Come mostrato in Figura 4, un utente, specializzato negli attori del sistema individuati in D1.6, interagisce con un qualsiasi browser web installato sul proprio dispositivo.

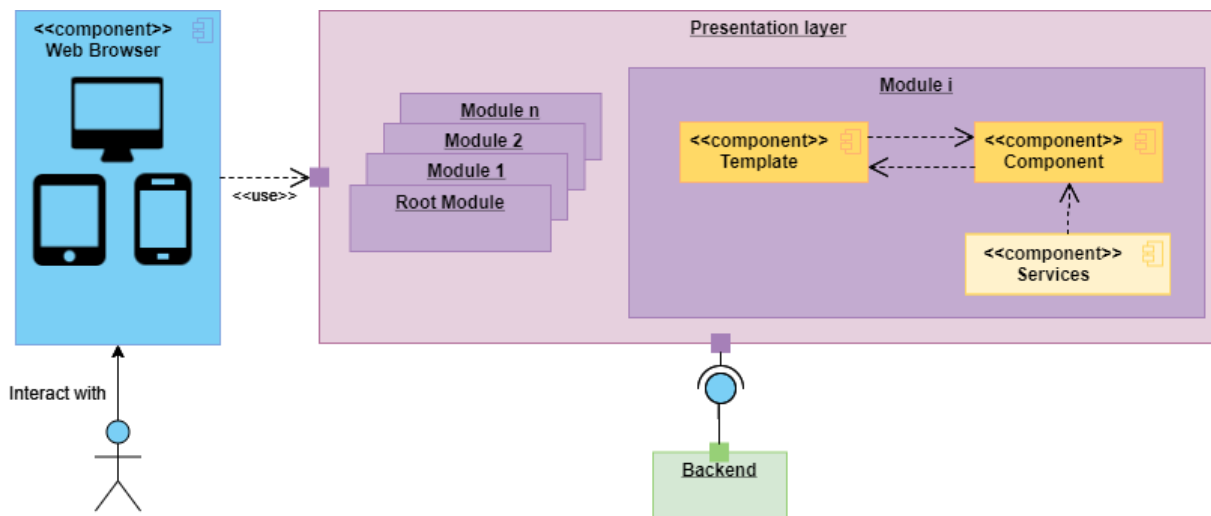


Figura 4. Presentation layer del software di monitoraggio in ATTICUS.

Il browser effettua una chiamata HTTPS alle componenti del *presentation layer* implementate in Angular. Gli elementi principali di una architettura Angular sono i moduli chiamati *NgModules* i quali

permettono la scrittura di codice modulare e ciascun modulo è dedicato ad uno specifico scopo logico. I moduli sono organizzati in maniera gerarchica con un modulo radice (Root Module). Le componenti insieme ai template sono compilati per ottenere il codice HTML e JavaScript da visualizzare. Ai servizi sono delegate le operazioni come l'implementazione di funzionalità comuni ai vari moduli e sono utilizzati per condividere informazioni tra classi che non si conoscono. La web-app comunica con il *backend*, il componente principale nell'*application layer*, per ottenere o aggiornare le informazioni da presentare all'utente. La richiesta viene eseguita su protocollo HTTPS, i dettagli sulle strategie di comunicazione sono illustrati nel dettaglio nell'architettura fisica. Nel *presentation layer* è implementato un client HTTPS che ha come obiettivi principali la capacità di effettuare richieste verso le API esposte dal *backend*, interpretare le risposte e aggiornare la visualizzazione della pagina.

2.1.3 Application layer

L'*application layer* è il livello responsabile di tutta la logica di business e come ruolo principale funge da ponte tra il *presentation layer* e il *database layer*; l'architettura logica del livello è illustrata in Figura 5 ed è valida sia per il software di monitoraggio sia per il DSS di livello 3.

L'architettura del *backend*, con implementazione basata sul framework Spring Boot, permette di individuare un componente Controller che riceve le richieste REST dal *presentation layer* ed effettua un mapping con il servizio che si occupa di gestire la richiesta. Il REST Controller definisce quali servizi corrispondono alla URL invocata e i parametri da leggere nella richiesta, mentre i servizi (*Services*) definiscono l'intera logica di business. Il framework Hibernate viene utilizzato per gestire l'accesso ai dati e fornisce una soluzione per separare la logica di business dall'accesso ai dati. Hibernate è un framework ORM (*Object Relational Mapping*) che si basa sulla tecnica di effettuare un mapping tra

le classi Java con le tabelle di un database, incapsulando le comuni operazioni SQL in modo da astrarre l'applicazione dal database sottostante e i dialetti SQL.

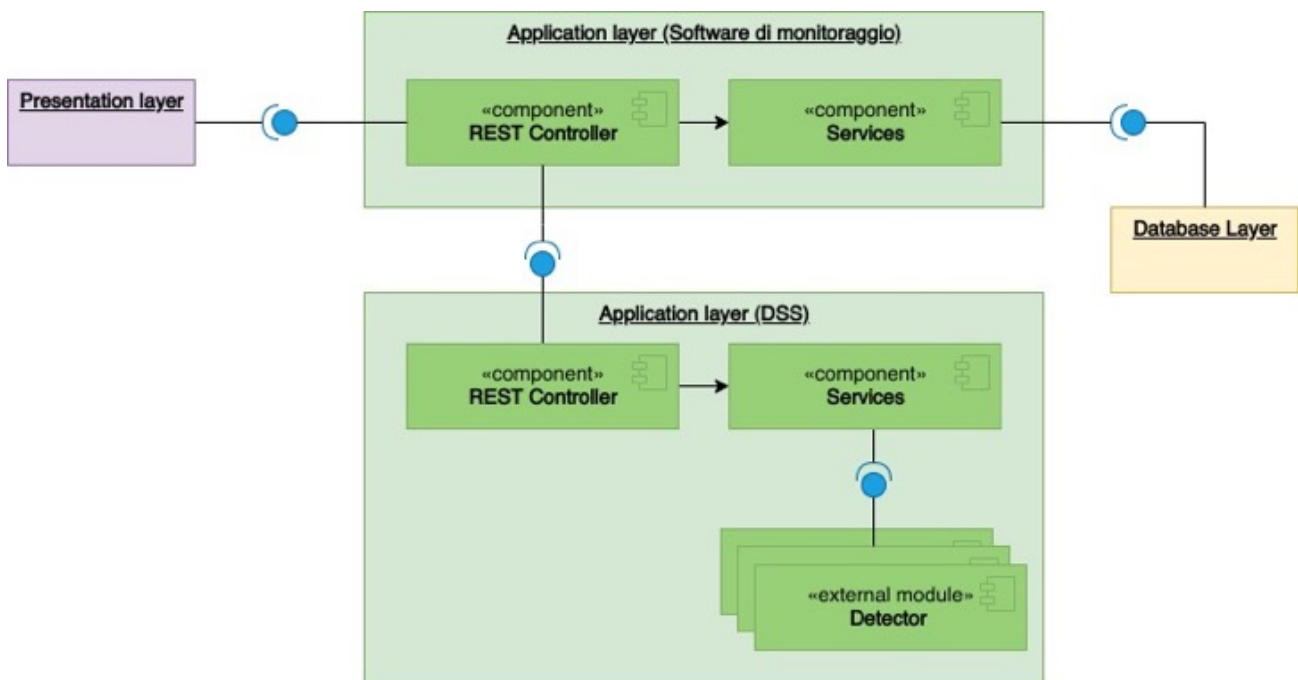


Figura 5. Application layer di software di monitoraggio e DSS di livello 3.

I principi alla base del framework Spring sono IoC (Inversion of Control) e DI (Dependency Injection). Secondo il primo principio il flusso di controllo è invertito, nel senso che, a differenza dell'approccio tradizionale nel quale lo sviluppatore controlla il flusso del programma, nell'IoC sono le sorgenti esterne (componenti, framework o servizi) a farlo. Il DI è un design pattern usato per implementare l'IoC. Tale principio permette di utilizzare i servizi in modo indipendente da chi li utilizza, ossia la connessione tra gli oggetti viene seguita da un assemblatore piuttosto che negli oggetti stessi. L'implementazione dei due concetti nel framework Spring permette di progettare soluzioni con basso accoppiamento, cambiare facilmente le implementazioni e aumentare la modularità e la testabilità del codice.

È bene notare che i *detector* implementati come moduli esterni, come precedentemente accennato, saranno sviluppati in Python e saranno completamente indipendenti dal DSS stesso. I detector esterni dovranno avere necessariamente in comune soltanto l'interfaccia che forniranno al DSS di livello 3. Questo approccio permette di avere diversi vantaggi: da un lato, qualora un detector dovesse smettere di funzionare (es: eccessivo uso di memoria causato da un bug), non ci sarebbero ripercussioni sul DSS di livello 3, che continuerebbe a funzionare normalmente. D'altro canto, è possibile sfruttare soluzioni già disponibili per l'analisi dei segnali e per il machine learning. I dettagli riguardanti tali aspetti sono riportati in D6.2.

Per quanto riguarda l'*application layer* del software di monitoraggio, sono previsti i seguenti sottosistemi:

- *gestione utenti*: si occupa della gestione di tutti gli utenti del sistema ATTICUS, ovvero Amministratore, Caregiver, Infermiere, Medico specialista, e Responsabile di Monitoraggio; il solo amministratore attraverso la funzionalità gestione utenti può inserire, ricercare attraverso dei filtri di ricerca, modificare ed eliminare un utente da ATTICUS.
- *gestione kit di monitoraggio*: si occupa della gestione dei kit di monitoraggio. Il solo utente con ruolo Amministratore di sistema può effettuare l'inserimento di un nuovo kit, ricercare un particolare kit di monitoraggio attraverso una sezione dedicata, per i soli kit attualmente non in uso può modificare ed eliminare particolari i kit di monitoraggio mentre per i kit in uso e assegnati ad un particolare assistito l'utente Amministratore di sistema potrà effettuare la sostituzione di un kit;
- *Arruolamento*: procedura sequenziale composta di 8 step di cui 5 obbligatori e 3 facoltativi. Il solo responsabile di monitoraggio, eseguiti gli step proposti dal sistema, rispettando opportune regole di validità, obbligatorio e univocità effettuerà a fine procedura l'arruolamento dell'assistito.

- *gestione assistiti*: si occupa di particolari operazioni che il medico specialista e il responsabile di monitoraggio possono eseguire sui dati amministrativi e clinici di un assistito permettendo di effettuare operazioni sull'assistito e sulla cartella clinica dell'assistito precedentemente scelto.
- *gestione allarmi*: si occupa della gestione degli allarmi generati.

2.1.4 Database layer

Il *database layer* è il livello responsabile dell'accesso ai dati e alla gestione della persistenza. Nel sistema ATTICUS, il software di monitoraggio utilizzerà un database relazionale (MySQL) per la gestione dei dati relativi alla parte gestionale del sistema; entrambi i sottosistemi, quindi software di monitoraggio e DSS di livello 3, utilizzeranno un database non-relazionale (MongoDB) per la gestione dei dati di monitoraggio degli assistiti. Si veda D6.1 per i dettagli relativi a tale scelta. Il collegamento dei database con il backend è previsto in doppia modalità lettura/scrittura: in un primo caso, dai database vengono estratte le informazioni necessarie all'*application layer*, mentre nel secondo caso i dati calcolati e prodotti nell'*application layer* devono essere salvati nella base dati.

Come mostrato in Figura 6, il *database layer* è ulteriormente decomposto in due livelli. Il primo livello SQL DATABASE ospita i database relazionali per i dati anagrafici e per i dati sanitari. Il secondo livello NOSQL DATABASE ospita il database documentale relativo alla memorizzazione dei dati di monitoraggio provenienti dai sensori.

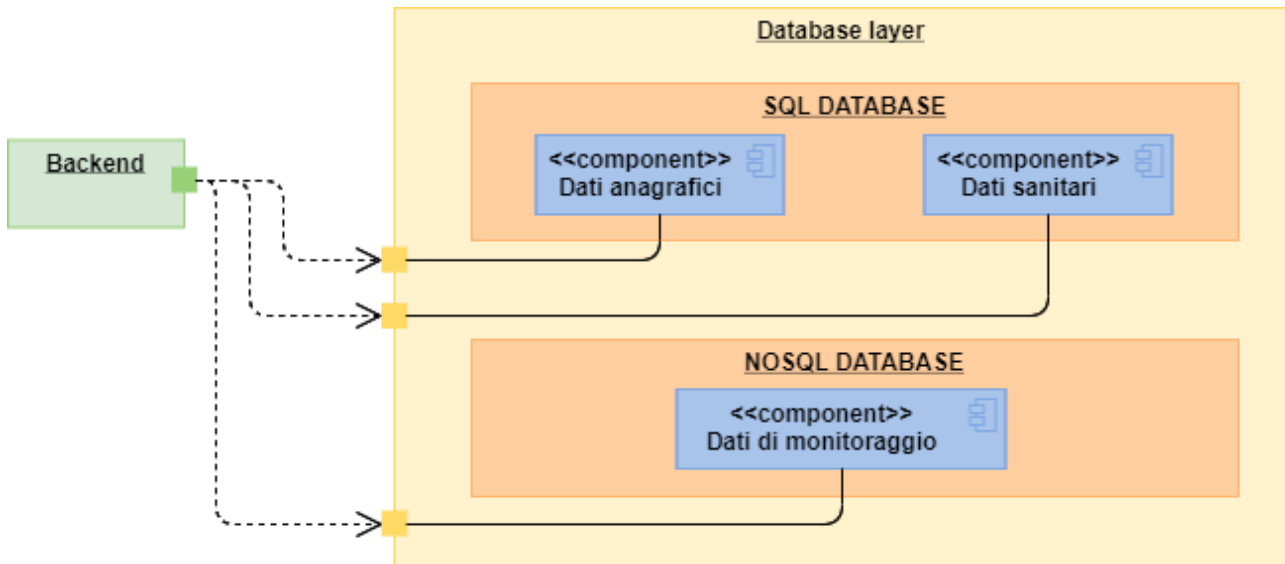


Figura 6. Database layer del sistema ATTICUS

I database relazionali salvano i dati in tabelle che modellano le diverse entità nel dominio dell'applicazione, una tabella ha un numero fisso di colonne e un numero indefinito di righe (*record*) e le colonne rappresentano i dati degli attributi dell'entità. Le tabelle possono avere delle colonne speciali come la chiave primaria (*primary key*) che contiene l'identificativo unico per ogni riga e le chiavi esterne (*foreign key*) che rappresentano un riferimento di una chiave primaria di un'altra riga di un'altra tabella. Tale collegamento tra due righe prende il nome di associazione, concetto alla base in questa tipologia di database.

Differentemente, i database NoSQL non seguono il modello relazionale descritto in precedenza. Tali database memorizzano i dati in collezioni (*collections*) e ogni collezione contiene i documenti. Le informazioni in questo contesto non vengono logicamente strutturate, ma sono semplicemente aggregate. Database di questo tipo sono privi della definizione di uno schema definito a priori (*schemaless*) che offre l'opportunità di scalarli orizzontalmente.

Data la natura intrinsecamente strutturata dei dati anagrafici e dei dati sanitari si è scelto di utilizzare database relazionali per la memorizzazione di queste informazioni a vantaggio della coerenza e dell'integrità. Per i dati di monitoraggio, invece, si è scelto l'utilizzo di database NoSQL in modo da gestire eventuali fonti eterogenee di dati che evolvono nel tempo, modellando l'informazione come documento senza la necessità di definire una struttura rigida dei contenuti.

2.2 Architettura fisica

La centrale di monitoraggio del sistema ATTICUS è definita secondo un modello distribuito, cioè un modello che si configura come un insieme di computer che, interconnessi attraverso una rete, collaborano all'esecuzione di diversi *task*. Sebbene i componenti della centrale di monitoraggio siano distribuiti attraverso diversi servizi in cloud, l'utente che accede al sistema non percepisce, durante il suo utilizzo, che si tratta di un sistema distribuito, dal momento che questo appare come un unico sistema centralizzato. Il modello distribuito ha il vantaggio di permettere una più agevole gestione degli errori e dei guasti, potendo isolarli e correggerli sulle singole componenti interessate e permettendo alle restanti componenti del sistema di continuare a funzionare correttamente; la *fault tolerance* rappresenta un'importante caratteristica del design della centrale di monitoraggio di ATTICUS, che mira a garantire un corretto funzionamento e una costante fornitura del servizio anche in caso di problemi di hardware, software e rete.

In questo paragrafo è illustrata la strategia di messa in opera (*deployment*) individuata per le due componenti software della centrale di monitoraggio, vale a dire software di monitoraggio e DSS di livello 3, con particolare riferimento all'architettura e servizi REST.

2.2.1 Strategia di deployment

Il modello distribuito individuato è basato sulla *Service-Oriented Architecture (SOA)*, tale modello è comunemente usato per progettare sistemi informatici basati sui servizi, ove un servizio può essere definito come una funzionalità atta a risolvere un problema. Essendo la SOA fondata sull'assunto che i problemi complessi possono essere risolti più efficacemente decomponendoli in una serie di problemi più piccoli e semplici, questo modello architetturale offre una molteplicità di servizi distinti l'uno dall'altro atti ad eseguire dei compiti per risolvere una serie di problemi. Tre sono i punti di forza di un SOA: la modularità, l'estendibilità e la riusabilità del codice. Le applicazioni possono essere costruite combinando diversi servizi creati completamente ex-novo, partendo da servizi già esistenti o modificando del codice già scritto e testato. Dato che il sistema di monitoraggio prevede la presenza di una web-app *standalone*, il modello SOA si presta efficacemente a questo caso.

Varie tecniche possono essere impiegate per programmare e costruire un'architettura SOA, una tra le più comuni è chiamata REST (REpresentational State Transfer). Alcune caratteristiche peculiari (dette vincoli) dell'approccio architetturale REST sono descritte in Tabella 1.

Tabella 1. Vincoli dell'approccio architetturale REST.

| Vincolo | Descrizione |
|-----------------------------------|--|
| Architettura client-server | È prevista un'architettura client-server, ove il client inoltra le richieste e il server risponde. Il client e il server possono operare indipendentemente l'uno dall'altro e non è previsto che ci sia alcuna dipendenza tra di loro. |
| Layered | Il sistema è suddiviso in vari <i>layer</i> per eseguire operazioni sulle risorse; un aspetto importante è che i <i>layer</i> possono essere aggiunti e rimossi all'occorrenza, sulla base delle esigenze client-server. |

| | |
|--------------------------|--|
| Cacheable | L'obiettivo del caching è di non generare mai la stessa risposta dal server al client al fine di velocizzare le comunicazioni e ridurre il peso. |
| Stateless | Il server non archivia alcuna informazione sullo stato del client. Quindi ogni richiesta dal client verso il server deve contenere tutte le informazioni necessarie per soddisfare la richiesta, non potendosi avvalere di dati archiviati in precedenza nella sessione. |
| Uniform interface | Dovrebbe esserci una modalità uniforme di interazione con il server, a prescindere dal tipo di applicazione client da cui parte la richiesta (che sia uno smartphone o un pc desktop). Questo vincolo pone una differenza fondamentale tra REST API e Non-REST API. |

Il design architetturale REST permette ai componenti del sistema di comunicare attraverso il protocollo HTTP. Si tratta di una comunicazione 'basata sulle risorse'; infatti, il *client* comunica al *server* l'azione che deve essere eseguita su una determinata risorsa mediante uno dei quattro verbi che contraddistinguono l'azione stessa da eseguire: *GET*, *PUT*, *POST* e *DELETE* (Figura 7). Lo scambio di comunicazioni avviene mediante file in formato JSON (JavaScript Object Notation), il quale è lo standard usato per la trasmissione asincrona dei dati tra client e server; il suo consolidato uso nelle SOA è legato al fatto che è un formato leggero, agevole da leggere e scrivere per le persone, inoltre è "language-agnostic", ovvero può essere usato con molteplici linguaggi di programmazione. La comunicazione dal client verso il server comprende sempre almeno il verbo che indica l'azione da eseguire e il percorso alla risorsa oggetto dell'azione.

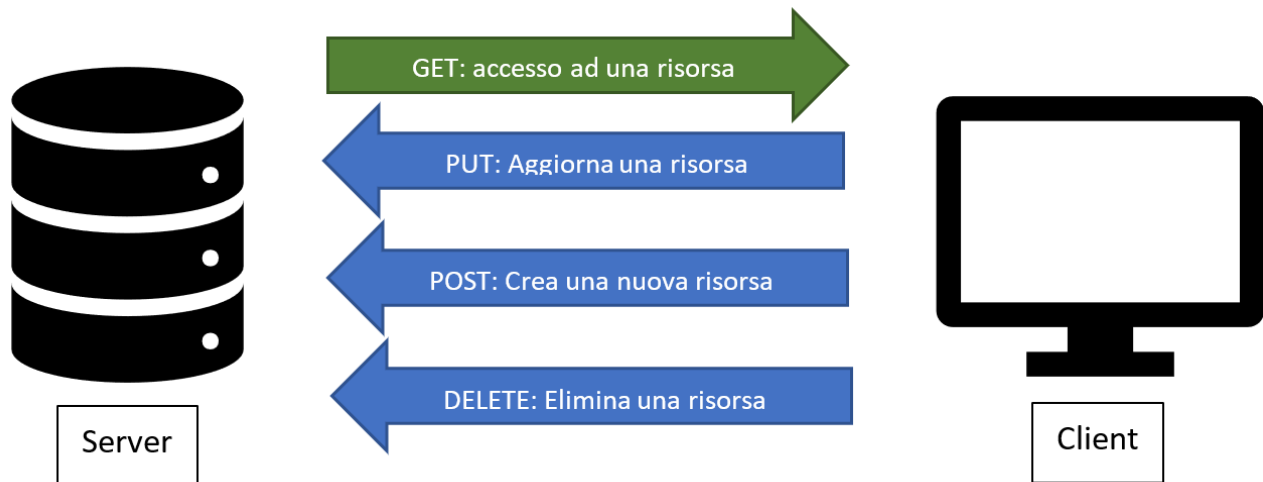


Figura 7. Azioni eseguibili sulle risorse dal sistema ATTICUS.

Quello di 'risorsa' è un altro concetto fondamentale su cui si basano i servizi REST: per risorsa si intende qualsiasi elemento oggetto di elaborazione su cui è possibile effettuare operazioni *GET*, *PUT*, *POST* e *DELETE*. Quindi, le risorse possono essere di vario tipo: foto, documenti, utenti, somministrazioni, etc. Ogni risorsa deve essere identificata univocamente attraverso l'*Uniform Resource Identifier* (URI). Le risorse del sistema ATTICUS rispondono ai seguenti standard di qualità previsti dalla letteratura attuale: sono usati esclusivamente sostantivi plurali per identificare le URI; l'azione *GET* non altera in alcun modo le risorse; le relazioni tra le risorse sono rappresentate da sotto-risorse nell'URI (es. *GET /users/UID/*).

I diversi servizi e componenti del sistema progettati sono distribuiti e gestiti attraverso containerizzazione. Questo approccio prevede la virtualizzazione delle risorse hardware eseguendo più istanze virtuali (chiamate *container*). La containerizzazione consente di isolare le applicazioni e i componenti di un sistema in più *container*, aumentando in tal modo anche il livello di sicurezza, dal momento che le informazioni di un'applicazione non sono liberamente accessibili da un'altra

applicazione. I *container* sono ambienti di esecuzione simili alle macchine virtuali, ovvero sono isolati dagli altri componenti che si trovano nella medesima infrastruttura e dispongono di una segregazione di filesystem, CPU, memoria, PID ecc. (Figura 8). Rispetto alle macchine virtuali, i *container* sono considerati più leggeri, dato che condividono il sistema operativo tra le applicazioni. Per la distribuzione e gestione dei *container* sarà usato un Kubernetes. Questo software open-source consente di gestire in modo scalabile i carichi di lavoro e i servizi containerizzati e permette di pianificare l'esecuzione di *container* in base alle risorse di elaborazione disponibili e a quelle richieste in ciascun *container*. Kubernetes analizza il traffico verso i server e il carico computazionale necessario per gestirlo, quindi orchestra e automatizza di conseguenza la distribuzione dei container. I container vengono eseguiti in raggruppamenti logici chiamati PoD (Point of Delivery); è possibile eseguire e ridimensionare uno o più container insieme come un solo PoD. Grazie a questo, si potranno avere più repliche di ogni PoD, in modo tale da rendere tutta l'infrastruttura scalabile oltre che, grazie all'aggregazione di uno o più nodi (i.e. server) dislocati gestiti sempre tramite Kubernetes, si potrà anche garantire l'alta disponibilità del software di monitoraggio e del DSS di livello 3.

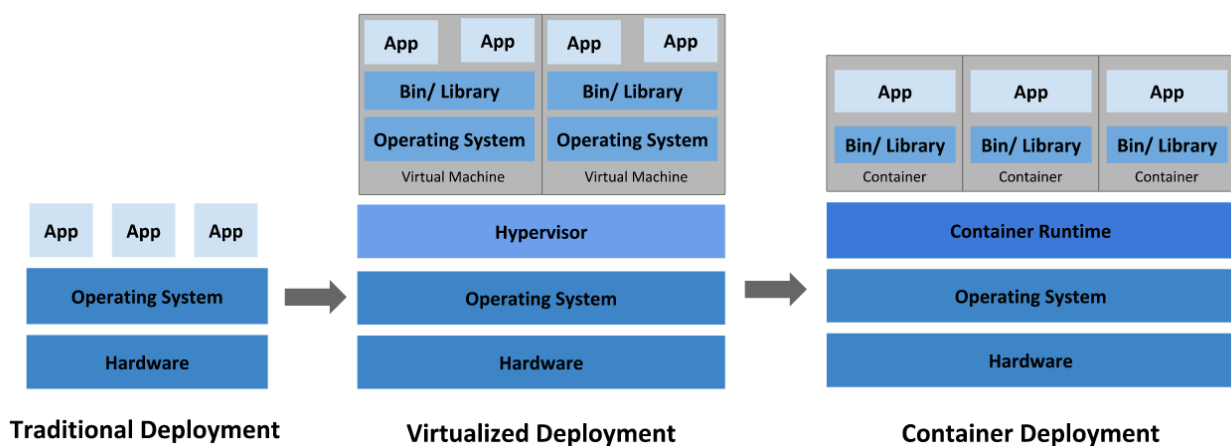


Figura 8. Evoluzione nella gestione delle infrastrutture informatiche (fonte: kubernetes.io).

2.2.2 Diagramma di deployment

Entrambi i sottosistemi che verranno installati nella centrale di monitoraggio, ovvero software di monitoraggio e DSS di livello 3, hanno in comune la distribuzione delle componenti attraverso i *container* Docker. Nello specifico, il software di monitoraggio è composto da due container: front-end (*presentation layer*) e back-end (*application layer*). Il DSS di livello 3 è composto da diversi container: sarà presente un *container* “core” (modulo principale) e un container per ogni modulo esterno del DSS. Sarà, poi, prevista la presenza di un container che fornisce i servizi del DBMS MySQL e uno per il DBMS MongoDB. Il diagramma di deployment complessivo della centrale di monitoraggio e dei due sottosistemi che la compongono è presentato in Figura 9.

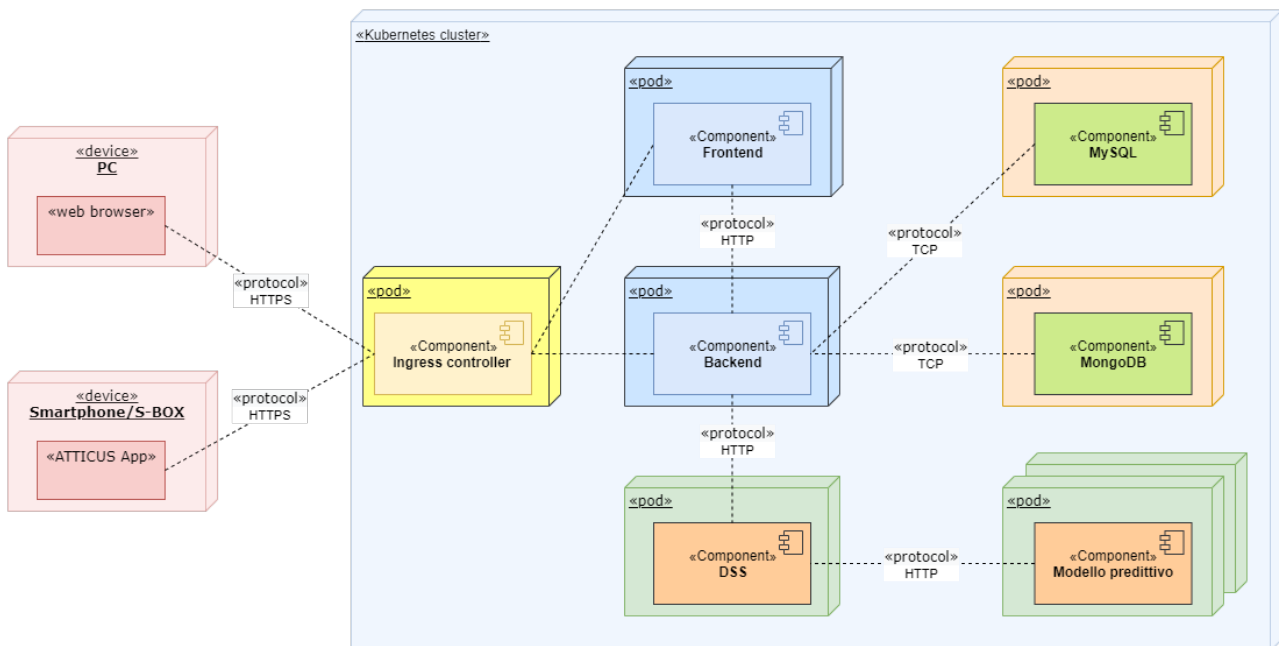


Figura 9. Diagramma di deployment del software di monitoraggio e del DSS di livello 3.

2.2.3 Configurazione di deployment

Il software di monitoraggio, come detto in precedenza, viene distribuito come immagini Docker e installato in container. Si utilizzerà Kubernetes come container orchestrator al fine di semplificare la gestione dei container. Come descritto in Figura 9, ogni componente (*front-end*, *back-end*, DSS e DBMS) verrà installata in un container differente fornito da Kubernetes (*PoD*). All'occorrenza si potrà decidere di aggiungere repliche dei container per garantire la scalabilità del sistema, oltre che l'alta disponibilità in caso di fallimenti. Per questo, verrà costituito nell'ambiente di produzione un cluster di due o più nodi: un nodo assumerà il ruolo di *master*, mentre i restanti avranno il ruolo di *worker*. Il nodo *master* gestirà i PoD che verranno istanziati sui *worker* e deciderà quando e come crearli. Nel caso uno dei worker dovesse andare offline, l'altro verrà usato come nodo di backup e, in questo modo, si garantirà *business continuity*.

Per la build e il rilascio dei container, invece, verrà utilizzato il sistema di *continuous integration (CI)* e *continuous deployment (CD)* di GitLab. Tale sistema prevede la configurazione di una *pipeline*, composta da diversi step (*stage*), che rappresentano le diverse fasi, come quella di *build* e di creazione di PoD su Kubernetes. L'intera *pipeline* verrà gestita da un *runner*, un container apposito in Kubernetes che avrà il compito di eseguire tutti i passi definiti nella *pipeline*. Ogni volta che inizierà una *build*, verrà creato un nuovo *runner*: in questo modo si potrà gestire efficientemente le *pipeline* delle diverse componenti di ATTICUS.

È prevista, inoltre, la presenza di un registry privato delle immagini Docker. Tale strumento risulta necessario affinché si possano mantenere diverse versioni del sistema pronte all'uso. Se ci sarà qualche problema con una nuova versione, sarà sempre possibile tornare alla versione precedente. Il deploy delle applicazioni su Kubernetes prevede la definizione di file di configurazione relativi al PoD, come le regole di forwarding del traffico, le porte da allocare e le risorse richieste. Per gestire

questi file in maniera semplificata, verrà utilizzato il gestore di pacchetti helm, che permette di gestire tutti i file di configurazione relativi ad uno specifico componente come un unico pacchetto.



Figura 10. Rappresentazione del processo di deployment tramite pipeline

Un pacchetto, quindi, rappresenterà una singola applicazione che verrà eseguita su Kubernetes. In Figura 10 è rappresentato tutto il processo di deployment tramite le pipeline di GitLab. Tutti i file di configurazione relativi alla build e al rilascio sono contenuti in un repository separato da quello in cui è contenuto il codice, per permettere maggiore flessibilità nel caso in cui fosse necessario modificare i parametri relativi al deployment. Nella fase di esecuzione dei tool di analisi statica, verrà sfruttata la suite offerta da SonarQube, che permette di rilevare bugs, code smells e vulnerabilità. Inoltre, offre un'interfaccia web dove è possibile consultare un insieme di grafici e statistiche relative ai dati forniti dai tool di analisi statica, ma anche la percentuale di coverage dei casi di test e uno storico delle stesse per avere una visione dello stato di salute dell'intero sistema a partire dalle fasi iniziali dello sviluppo. Il monitoraggio dei servizi deployati verrà affidato al *Monitoring Stack* formato da un aggregatore di metriche (Prometheus), un aggregatore di file di log (FluentBit), un analizzatore di logs (Loki) e una interfaccia grafica (Grafana) per facilitare l'esplorazione di questi dati. Nella

Tabella 2, vengono riportati i software utilizzati per il deployment con le relative versioni installate e le porte utilizzate.

Tabella 2. Versione del software utilizzato per il deploy

| Nome | Versione | Descrizione |
|--------------------|------------|---|
| Kubernetes | 1.18.3 | Sistema di gestione di cluster e orchestrazione di container |
| Helm | 3.2.1 | Strumento per l'installazione e la gestione del ciclo di vita di applicazioni su Kubernetes |
| Docker | 19.03.8 | Tecnologia di containerizzazione |
| Gitlab | 12.10.2-ee | Piattaforma di gestione per repository Git, con supporto a CI/CD |
| MySQL | 8.0.22 | RDBMS |
| MongoDB | 4.2.7 | DBMS non relazionale |
| Sonarqube | 8.6 | Piattaforma per il controllo della qualità del codice mediante analisi statica |
| PHPMYAdmin | 5.0.2 | Piattaforma per la gestione e amministrazione di database MySQL |
| grafana/loki-stack | 2.4.1 | Monitoring stack per l'analisi e la visualizzazione di metriche e log files relativi al sistema |
| CentOS | 7.8.2003 | Distribuzione linux enterprise per macchine server |

2.2.4 Continuous machine learning

Il DSS di livello 3 utilizza moduli esterni per implementare i *detector*, i quali sfruttano modelli predittivi specializzati per la rilevazione di specifiche anomalie. Dato che in ATTICUS è prevista anche la taggatura e l'aggiornamento dei modelli alla base dei detector tramite nuovi dati (es. segnali ECG), i suddetti modelli predittivi prevederanno un sistema di aggiornamento continuo dei modelli utilizzati con i nuovi dati. Per fare ciò, bisogna (i) tracciare le diverse versioni dei modelli e dei dati utilizzati da essi e (ii) costruire una *pipeline* di *continuous machine learning* per l'addestramento, la verifica e il rilascio dei modelli in maniera automatizzata. Tutto questo verrà gestito tramite le *pipeline* di *continuous integration* (CI) di GitLab precedentemente descritte, quindi le stesse che

vengono usate anche per il rilascio delle altre componenti del software di monitoraggio. Ad ogni detector saranno associati tre differenti repository (Figura 11):

- **Repository base:** comune a tutti i modelli predittivi, conterrà il codice per il caricamento del modello e l'esposizione delle API di predizione. In questo modo, si garantisce un'interfaccia comune a tutti i modelli;
- **Repository dati:** conterrà i dati utilizzati per l'addestramento e la verifica del modello, insieme agli script che eseguono queste due operazioni;
- **Repository di deploy:** conterrà il codice per la procedura di lancio del training e rilascio del modello (*pipeline*).

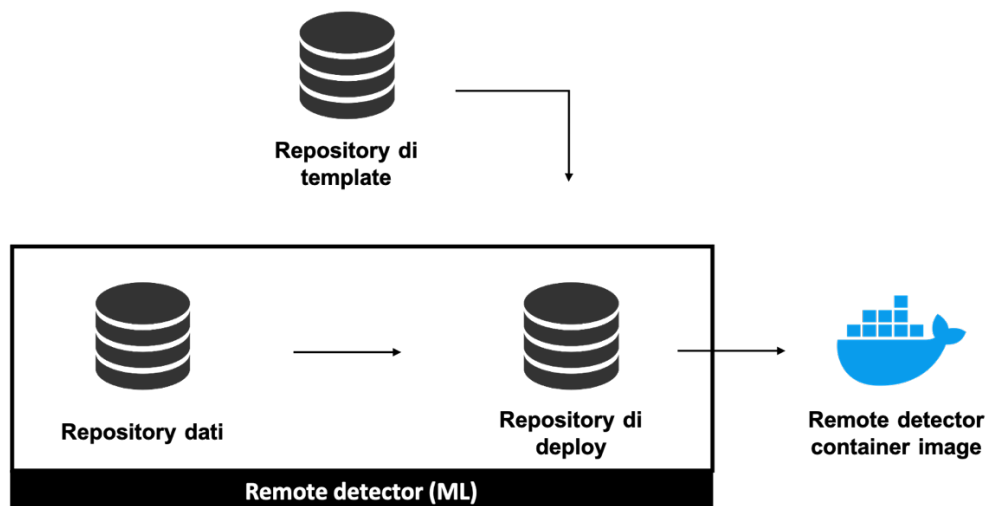


Figura 11. Schema dei repository usati dai detector basati su machine learning.

Il repository dati verrà aggiornato di volta in volta con i nuovi dati. Una volta finito l'addestramento, verrà fatta la validazione e verranno misurate le prestazioni attraverso le seguenti metriche:

- **Accuratezza:** calcolata come numero di predizioni corrette su numero di predizioni totali;

- **AUC:** l'Area Under the ROC Curve permette di avere un'idea di quanto un classificatore è in grado di distinguere le classi del problema. L'AUC va calcolata per ogni classe: in questo caso, ci si riferisce alla media delle AUC calcolata su tutte le classi.

I nuovi dati verranno integrati nel nuovo modello del detector solo se le prestazioni dello stesso non diminuiscono. L'aggiornamento con nuovi dati farà partire la *pipeline* contenuta nel repository di *deploy* che, scaricati i dati e il codice per l'esposizione delle API, procederà ad addestrare e verificare il nuovo modello. Infine, se il nuovo modello riporta prestazioni peggiori rispetto al precedente, il detector verrà rilasciato su un nuovo PoD nel cluster. Kubernetes si occuperà di sostituire il PoD contenente il vecchio detector con quello nuovo, facendo confluire verso questo le nuove richieste che arriveranno e distruggendo quello vecchio quando sarà completamente scarico. Questo garantirà l'assenza completa di *downtime* durante gli aggiornamenti. Se il modello non rispetterà il criterio di aggiornamento precedentemente descritto, la pipeline verrà interrotta e il vecchio detector continuerà a funzionare. Uno schema del processo è rappresentato in Figura 12.

Dato che i tre repository di ogni detector saranno collegati tra loro, l'aggiornamento di ognuno di essi lancerà una nuova pipeline che aggiornerà il modello attualmente in uso. Quindi, anche una variazione nel codice di esposizione delle API di predizione, ad esempio, farà partire la pipeline di tutti i detector, rilasciandone una nuova versione aggiornata.

L'aggiornamento del repository dati verrà gestito da un servizio che periodicamente controllerà la disponibilità di nuovi dati etichettati. Una volta raggiunta una certa quantità di nuovi dati, verrà aggiornato il repository e, quindi, lanciata una nuova pipeline.

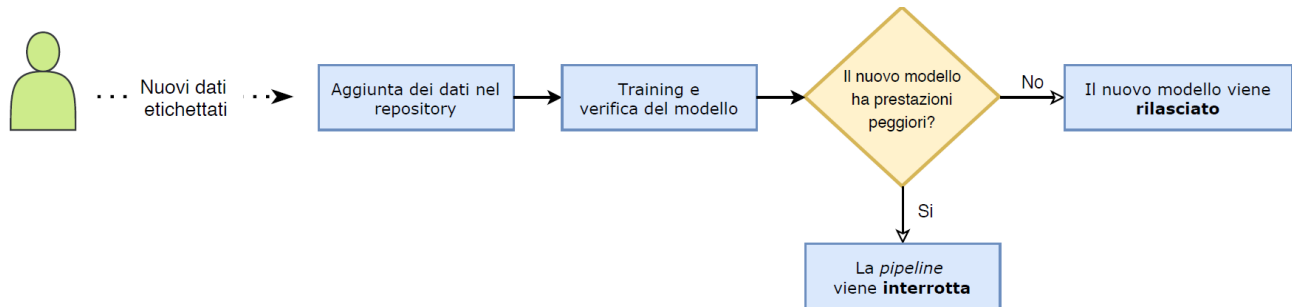


Figura 12. Schema del processo di aggiornamento dei modelli predittivi.

3 Servizi del software di monitoraggio

In questa sezione si elencano i servizi del software di monitoraggio, divisi per i diversi sottosistemi individuati nella sezione precedente.

3.1 Servizi gestione utenti

| | |
|--|--|
| Nome | getUser |
| URL | GET - /atticus/api/user/getUser/{id} |
| Recupera le informazioni dell'Utente per il quale si desidera ottenere informazioni. Viene dato in input l'id Utente | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce l'Utente: return Optional<UserDTO></p> <p>Esempio</p> <p>Esempio: /atticus/api/user/getUser/2</p> <pre>{ "id": 1, "idRole": 1, "idResidenceCity": 5648, "ssn": "RSSMRA65A01B519L", "surname": "Rossi", "name": "Mariano",</pre> |

```

"sex": "M",
"birthDate": -157766400000,
"idBirthPlace": 5670,
"birthState": "IT",
"abroadBirthPlace": null,
"residenceAddress": "berlinguer",
"residencePc": "65014",
"cellNumber": "3123434000",
"email": "mario@rossi.it",
"suspended": "0",
"doctorCode": null,
"cityByIdResidenceCity": {
  "id": 5648,
  "name": "San Salvo",
  "province": "Chieti",
  "provinceAbbreviation": "CH"
},
"cityByIdBirthPlace": {
  "id": 5670,
  "name": "Acquaviva Collecroce",
  "province": "Campobasso",
  "provinceAbbreviation": "CB"
}
}

```

| | |
|------------|---|
| STATUS 500 | <p>AtticusException: UserException</p> <ul style="list-style-type: none"> Nel caso in cui l'utente non è presente all'interno de sistema: "Attenzione! L'Utente non esiste in Atticus" |
|------------|---|

| | |
|---|---|
| Nome | findUsersWithPagination |
| URL | POST - /atticus/api/user/findUsersWithPagination/{allUser} |
| Recupera la lista paginata di tutti gli utenti presenti in Atticus o degli utenti richiesti in base ai filtri inseriti. | |
| Risposta | |
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "id" }, "filters": { "surname": "Rossi", "name": "", "doctorCode": "", "ssn": null, } }</pre> |

| | |
|-------------------|---|
| | <pre> } } </pre> |
| <p>STATUS 200</p> | <p>Esempio: application/json</p> <p>Restituisce la lista paginata degli Utenti: return Page<UserDTO></p> <pre> { "content": [{ "id": 1, "idRole": 1, "idResidenceCity": 5648, "ssn": "RSSMRA65A01B519L", "surname": "Rossi", "name": "Mariano", "sex": "M", "birthDate": -157766400000, "idBirthPlace": 5670, "birthState": "IT", "abroadBirthPlace": null, "residenceAddress": "berlinguer", "residencePc": "65014", "cellNumber": "3123434000", }] } </pre> |

| | |
|--|---|
| | <pre> "email": "mario@rossi.it", "suspended": "0", "doctorCode": null, "cityByldResidenceCity": { "id": 5648, "name": "San Salvo", "province": "Chieti", "provinceAbbreviation": "CH" }, "cityByldBirthPlace": { "id": 5670, "name": "Acquaviva Collecroce", "province": "Campobasso", "provinceAbbreviation": "CB" } }, { "id": 2, "idRole": 5, "idResidenceCity": 5670, "ssn": "RSSFRC72L53B519U", "surname": "Rossi", "name": "Federica", "sex": "F", </pre> |
|--|---|

| | |
|--|--|
| | <pre> "birthDate": 80352000000, "idBirthPlace": 5670, "birthState": "IT", "abroadBirthPlace": null, "residenceAddress": "via XXIV Maggio", "residencePc": "86100", "cellNumber": "3333434000", "email": "federica@rossi.it", "suspended": "0", "doctorCode": "1111", "cityByldResidenceCity": { "id": 5670, "name": "Acquaviva Collecroce", "province": "Campobasso", "provinceAbbreviation": "CB" }, "cityByldBirthPlace": { "id": 5670, "name": "Acquaviva Collecroce", "province": "Campobasso", "provinceAbbreviation": "CB" } }], </pre> |
|--|--|

```
"pageable": {  
  "sort": {  
    "sorted": true,  
    "unsorted": false,  
    "empty": false  
  },  
  "offset": 0,  
  "pageNumber": 0,  
  "pageSize": 5,  
  "paged": true,  
  "unpaged": false  
},  
"totalPages": 1,  
"last": true,  
"totalElements": 2,  
"size": 5,  
"number": 0,  
"sort": {  
  "sorted": true,  
  "unsorted": false,  
  "empty": false  
},  
"numberOfElements": 2,  
"first": true,
```

| | |
|------------|---|
| | <pre>"empty": false }</pre> |
| STATUS 500 | <p>AtticusException: UserException</p> <ul style="list-style-type: none"> • Se non è stato inserito alcun filtro e si seleziona il bottone Filtra: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” con la lista dei campi vuoti e un messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente e si seleziona il bottone Filtra: “Attenzione! Campi non popolati correttamente” con la lista dei campi non validi e messaggio “Campo non valido” e campo che ha generato l’errore <p>Esempio di eccezione:</p> <pre>{ "errorCode": "SEARCH_USER_FILTER_ERROR", "message": "Attenzione! Campi non popolati correttamente", "fieldsError": [{ "code": "SEARCH_USER_INCORRECT_ID_ROLE", "item": "-7", "field": "idRoleUserFilter", "label": "Campo non valido" }] }</pre> |

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|---|
| | } |
|--|---|

| | |
|--|--|
| Nome | addNewUser |
| URL | POST -/atticus/api/user/addNewUser |
| <p>Permette di aggiungere un nuovo utente in Atticus, solo dopo aver superato, per i campi richiesti, opportuni controlli di univocità, obbligatorietà e validità dei campi.</p> | |
| Input | <pre>{ "surname": "Genovese", "name": "Federica", "ssn": "GNVFRC94S65B519P", "sex": "F", "birthDate": "1994-11-25", "birthState": "IT", "idBirthPlace": 5675, "idResidenceCity": 5648, "residenceAddress": "via Berlinguer", "residencePc": "15066", "cellNumber": "3331122331", "email": "fede.genovese@gmail.com", "idRole": 2, "doctorCode": null }</pre> |

40

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| Risposta | |
|------------|---|
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo del salvataggio a database: return ResponseOperation</p> <pre>{ "success": true }</pre> |
| STATUS 500 | <p>AtticusException: UserException</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore • Se uno dei campi obbligatori non è popolato e uno non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio”/“Campo non valido” e campo che ha generato l’errore <p>Esempio di eccezione:</p> <pre>{ "errorCode": "NEW_USER_ERROR",</pre> |

| | |
|--|---|
| | <pre>"message": "Attenzione! Campi obbligatori non popolati o non popolati correttamente", "fieldsError": [{ "code": "USER_EMPTY_SURNAME", "item": "", "field": "surnameUser", "label": "Campo obbligatorio" }, { "code": "USER_INCORRECT_GENDER", "item": "Me", "field": "genderUser", "label": "Campo non valido" }] }</pre> <ul style="list-style-type: none">• Se il codice fiscale o l'email inseriti sono già presenti all'interno del sistema: "Attenzione! L'Utente già presente in Atticus" ed il campo che ha generato l'errore <pre>{ "errorCode": "ADD_USER_EXISTS", "message": "Attenzione! Utente già presente", "fieldsError": [</pre> |
|--|---|

| | |
|--|---|
| | <pre> { "code": "USER_SSN_ALREADY_EXIST", "item": null, "field": "ssnUser", "label": " " }] } </pre> |
|--|---|

| | |
|---|--|
| Nome | updateUser |
| URL | PUT -/atticus/api/user/updateUser |
| Aggiorna l'utente modificato, dopo una serie di controlli di veridicità, obbligatorietà e unicità | |
| Input | <pre> { "id": 10, "surname": "Genovese", "name": "Federica", "ssn": "GNVFRC94S65B519P", "sex": "F", "birthDate": "1994-11-25", "birthState": "IT", "idBirthPlace": 5675, "idResidenceCity": 5648, </pre> |

| | |
|------------|--|
| | <pre> "residenceAddress": "via Berlinguer", "residencePc": "15066", "cellNumber": "3331122331", "email": "fede.genovese@gmail.com", "idRole": 2, "doctorCode": null } </pre> |
| Risposta | |
| STATUS 200 | <pre> application/json Ritorna il successo del salvataggio nel database: return ResponseOperation Esempio: /atticus/api/user/updateUser { "success": true } </pre> |
| STATUS 500 | <p>Atticus Exception: UserException</p> <ul style="list-style-type: none"> Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e la lista dei campi vuoti con messaggio “Campo obbligatorio” e campo che ha generato l’errore |

- Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore
- Se uno dei campi obbligatori non è popolato e uno non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio”/“Campo non valido” e campo che ha generato l’errore

Esempio di eccezione:

```
{  
  "errorCode": "UPDATE_USER_ERROR",  
  "message": "Attenzione! Campi obbligatori non popolati o non popolati  
correttamente",  
  "fieldsError": [  
    {  
      "code": "USER_EMPTY_SURNAME",  
      "item": "",  
      "field": "surnameUser",  
      "label": "Campo obbligatorio"  
    },  
    {  
      "code": "USER_INCORRECT_SSN",  
      "item": "HUEKN/(&",
```

| | |
|--|---|
| | <pre> "field": "ssnUser", "label": "Campo non valido" }] } </pre> <ul style="list-style-type: none"> Se il codice fiscale o l'email inseriti sono già presenti all'interno del sistema: "Attenzione! L'Utente già presente in Atticus" e il campo che ha generato l'errore <pre> { "errorCode": "ADD_USER_EXISTS", "message": "Attenzione! Utente già presente", "fieldsError": [{ "code": "USER_EMAIL_ALREADY_EXIST", "item": null, "field": "emailUser", "label": " " }] } </pre> <ul style="list-style-type: none"> Se l'utente non è abilitato alla modifica del ruolo e del codice medico: "Attenzione! Impossibile modificare il Ruolo e il Codice Medico" |
|--|---|

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|--|
| | dell'Utente selezionato, perché quest'ultimo ha assistiti ad esso associati” |
|--|--|

| | |
|--|---|
| Nome | Delete |
| URL | DELETE - /atticus/api/user/deleteUser/{id} |
| Elimina dal database l'utente che si desidera eliminare. L'eliminazione avviene in base all'id utente passato. | |
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "id" }, "filters": { "surname": "Rossi", "name": "", "doctorCode": "", "ssn": null, } }</pre> |
| Risposta | |

47

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| | |
|------------|---|
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo dell'eliminazione nel database: return ResponseOperation</p> <p>Esempio: /atticus/api/user/delete/1</p> <pre>{ "success": true }</pre> |
| STATUS 500 | <p>AtticusException: UserException</p> <ul style="list-style-type: none"> Se l'utente non è presente all'interno del sistema: "Attenzione! L'Utente non esiste in Atticus" |

| | |
|--|---|
| Nome | getRole |
| URL | GET - http://localhost:9090/atticus/api/role/getRole |
| <p>Recupera la lista di tutti i ruoli utente presenti nel database e le rispettive abilitazioni alle funzionalità del sistema.</p> | |
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "id" }, }</pre> |

| | |
|-------------------|--|
| | <pre> "filters": { "surname": "Rossi", "name": "", "doctorCode": "", "ssn": null, } </pre> |
| <p>Risposta</p> | |
| <p>STATUS 200</p> | <pre> application/json Restituisce la lista dei ruoli: return List<RoleDTO> [{ "id": 1, "name": "Amministratore", "featureToRolesById": [{ "idRole": 1, "idFeature": 1, "featureByIdFeature": { "id": 1, "name": "gestione_utenti" } }] },] </pre> |

```
{
  "idRole": 1,
  "idFeature": 2,
  "featureByIdFeature": {
    "id": 2,
    "name": "nuovo_utente"
  }
},
{
  "idRole": 1,
  "idFeature": 3,
  "featureByIdFeature": {
    "id": 3,
    "name": "lista_utenti"
  }
},
{
  "idRole": 1,
  "idFeature": 4,
  "featureByIdFeature": {
    "id": 4,
    "name": "dettaglio_utente"
  }
},
}
```

```
{
  "idRole": 1,
  "idFeature": 5,
  "featureByIdFeature": {
    "id": 5,
    "name": "modifica_utente"
  }
},
{
  "idRole": 1,
  "idFeature": 6,
  "featureByIdFeature": {
    "id": 6,
    "name": "elimina_utente"
  }
},
{
  "idRole": 1,
  "idFeature": 7,
  "featureByIdFeature": {
    "id": 7,
    "name": "gestione_kit_monitoraggio"
  }
},
}
```

```
{
  "idRole": 1,
  "idFeature": 8,
  "featureByIdFeature": {
    "id": 8,
    "name": "nuovo_kit_monitoraggio"
  }
},
{
  "idRole": 1,
  "idFeature": 9,
  "featureByIdFeature": {
    "id": 9,
    "name": "lista_kit_monitoraggio"
  }
},
{
  "idRole": 1,
  "idFeature": 10,
  "featureByIdFeature": {
    "id": 10,
    "name": "dettaglio_kit_monitoraggio"
  }
},
}
```

```
{
  "idRole": 1,
  "idFeature": 11,
  "featureByIdFeature": {
    "id": 11,
    "name": "modifica_kit_monitoraggio"
  }
},
{
  "idRole": 1,
  "idFeature": 12,
  "featureByIdFeature": {
    "id": 12,
    "name": "elimina_kit_monitoraggio"
  }
},
{
  "idRole": 1,
  "idFeature": 13,
  "featureByIdFeature": {
    "id": 13,
    "name": "sostituisci_kit_monitoraggio"
  }
}
}
```

```
]
},
{
  "id": 2,
  "name": "Caregiver",
  "featureToRolesById": [
    {
      "idRole": 2,
      "idFeature": 15,
      "featureByIdFeature": {
        "id": 15,
        "name": "lista_assistiti"
      }
    },
    {
      "idRole": 2,
      "idFeature": 18,
      "featureByIdFeature": {
        "id": 18,
        "name": "visualizza_scheda_assistito"
      }
    }
  ]
},
]
```

```
{
  "id": 3,
  "name": "Infermiere",
  "featureToRolesById": []
},
{
  "id": 4,
  "name": "Medico Specialista",
  "featureToRolesById": [
    {
      "idRole": 4,
      "idFeature": 16,
      "featureByIdFeature": {
        "id": 16,
        "name": "attiva_disattiva_monitoraggio_assistito"
      }
    },
    {
      "idRole": 4,
      "idFeature": 18,
      "featureByIdFeature": {
        "id": 18,
        "name": "visualizza_scheda_assistito"
      }
    }
  ]
}
```

```
},  
{  
  "idRole": 4,  
  "idFeature": 20,  
  "featureByIdFeature": {  
    "id": 20,  
    "name": "gestione_follow_up"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 21,  
  "featureByIdFeature": {  
    "id": 21,  
    "name": "nuovo_follow_up"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 22,  
  "featureByIdFeature": {  
    "id": 22,  
    "name": "storico_follow_up"  
  }  
}
```

```
},  
{  
  "idRole": 4,  
  "idFeature": 23,  
  "featureByIdFeature": {  
    "id": 23,  
    "name": "soglie_allarme"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 24,  
  "featureByIdFeature": {  
    "id": 24,  
    "name": "gestione_esami_strumentali"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 25,  
  "featureByIdFeature": {  
    "id": 25,  
    "name": "nuovo_esame_strumentale"  
  }  
}
```

```

    },
    {
      "idRole": 4,
      "idFeature": 26,
      "featureByIdFeature": {
        "id": 26,
        "name": "storico_esami_strumentali"
      }
    },
    {
      "idRole": 4,
      "idFeature": 27,
      "featureByIdFeature": {
        "id": 27,
        "name": "confronta_esami_strumentali"
      }
    },
    {
      "idRole": 4,
      "idFeature": 28,
      "featureByIdFeature": {
        "id": 28,
        "name": "gestione_esami_ematochimici"
      }
    }
  
```

```
},  
{  
  "idRole": 4,  
  "idFeature": 29,  
  "featureByIdFeature": {  
    "id": 29,  
    "name": "nuovo_esame_ematochimico"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 30,  
  "featureByIdFeature": {  
    "id": 30,  
    "name": "storico_esami_ematochimici"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 31,  
  "featureByIdFeature": {  
    "id": 31,  
    "name": "confronta_esami_ematochimici"  
  }  
}
```

```
},  
{  
  "idRole": 4,  
  "idFeature": 32,  
  "featureByIdFeature": {  
    "id": 32,  
    "name": "gestione_terapia"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 33,  
  "featureByIdFeature": {  
    "id": 33,  
    "name": "nuova_terapia"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 34,  
  "featureByIdFeature": {  
    "id": 34,  
    "name": "storico_terapia"  
  }  
}
```

```
},  
{  
  "idRole": 4,  
  "idFeature": 35,  
  "featureByIdFeature": {  
    "id": 35,  
    "name": "gestione_appuntamenti"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 36,  
  "featureByIdFeature": {  
    "id": 36,  
    "name": "nuovo_appuntamento"  
  }  
},  
{  
  "idRole": 4,  
  "idFeature": 37,  
  "featureByIdFeature": {  
    "id": 37,  
    "name": "storico_appuntamento"  
  }  
}
```

```
    },  
    {  
      "idRole": 4,  
      "idFeature": 38,  
      "featureByIdFeature": {  
        "id": 38,  
        "name": "assistito_assistenza_domiciliare"  
      }  
    }  
  ]  
},  
{  
  "id": 5,  
  "name": "Responsabile di Monitoraggio",  
  "featureToRolesById": [  
    {  
      "idRole": 5,  
      "idFeature": 14,  
      "featureByIdFeature": {  
        "id": 14,  
        "name": "arruolamento"  
      }  
    }  
  ],  
  {
```

```
    "idRole": 5,  
    "idFeature": 15,  
    "featureByIdFeature": {  
      "id": 15,  
      "name": "lista_assistiti"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 16,  
    "featureByIdFeature": {  
      "id": 16,  
      "name": "attiva_disattiva_monitoraggio_assistito"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 17,  
    "featureByIdFeature": {  
      "id": 17,  
      "name": "elimina_assistito"  
    }  
  },  
  {
```

```
    "idRole": 5,  
    "idFeature": 18,  
    "featureByIdFeature": {  
      "id": 18,  
      "name": "visualizza_scheda_assistito"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 19,  
    "featureByIdFeature": {  
      "id": 19,  
      "name": "modifica_dati_amministrativi"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 20,  
    "featureByIdFeature": {  
      "id": 20,  
      "name": "gestione_follow_up"  
    }  
  },  
  {
```

```
    "idRole": 5,  
    "idFeature": 21,  
    "featureByIdFeature": {  
      "id": 21,  
      "name": "nuovo_follow_up"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 22,  
    "featureByIdFeature": {  
      "id": 22,  
      "name": "storico_follow_up"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 23,  
    "featureByIdFeature": {  
      "id": 23,  
      "name": "soglie_allarme"  
    }  
  },  
  {
```

```
    "idRole": 5,  
    "idFeature": 24,  
    "featureByIdFeature": {  
      "id": 24,  
      "name": "gestione_esami_strumentali"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 25,  
    "featureByIdFeature": {  
      "id": 25,  
      "name": "nuovo_esame_strumentale"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 26,  
    "featureByIdFeature": {  
      "id": 26,  
      "name": "storico_esami_strumentali"  
    }  
  },  
  {
```

```
    "idRole": 5,  
    "idFeature": 27,  
    "featureByIdFeature": {  
      "id": 27,  
      "name": "confronta_esami_strumentali"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 28,  
    "featureByIdFeature": {  
      "id": 28,  
      "name": "gestione_esami_ematochimici"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 29,  
    "featureByIdFeature": {  
      "id": 29,  
      "name": "nuovo_esame_ematochimico"  
    }  
  },  
  {
```

```
    "idRole": 5,
    "idFeature": 30,
    "featureByIdFeature": {
      "id": 30,
      "name": "storico_esami_ematochimici"
    }
  },
  {
    "idRole": 5,
    "idFeature": 31,
    "featureByIdFeature": {
      "id": 31,
      "name": "confronta_esami_ematochimici"
    }
  },
  {
    "idRole": 5,
    "idFeature": 32,
    "featureByIdFeature": {
      "id": 32,
      "name": "gestione_terapia"
    }
  },
  {
```

```
    "idRole": 5,  
    "idFeature": 33,  
    "featureByIdFeature": {  
      "id": 33,  
      "name": "nuova_terapia"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 34,  
    "featureByIdFeature": {  
      "id": 34,  
      "name": "storico_terapia"  
    }  
  },  
  {  
    "idRole": 5,  
    "idFeature": 35,  
    "featureByIdFeature": {  
      "id": 35,  
      "name": "gestione_appuntamenti"  
    }  
  },  
  {
```

```
    "idRole": 5,
    "idFeature": 36,
    "featureByIdFeature": {
      "id": 36,
      "name": "nuovo_appuntamento"
    }
  },
  {
    "idRole": 5,
    "idFeature": 37,
    "featureByIdFeature": {
      "id": 37,
      "name": "storico_appuntamento"
    }
  },
  {
    "idRole": 5,
    "idFeature": 38,
    "featureByIdFeature": {
      "id": 38,
      "name": "assistito_assistenza_domiciliare"
    }
  }
}
```

| | |
|--|------------------------------------|
| | <pre> }] </pre> |
|--|------------------------------------|

3.2 Servizi gestione kit di monitoraggio

| | |
|--|--|
| Nome | getAvailableMonitoringKits |
| URL | GET - /atticus/api/monitoringKit/getAvailableMonitoringKits |
| Recupera i Kit di Monitoraggio con stato uguale a "D" (Disponibile). | |
| Risposta | |
| STATUS 200 | <pre> application/json return List<MonitoringKitDTO> [{ "id": 3, "code": "22:22:22", "kitTypeId": 1, "state": "D", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", } }] </pre> |

| | |
|--|---|
| | <pre> "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }, "ssnPerson": null, "personByMonitoringKit": null, "monitoringKitList": null }] </pre> |
|--|---|

| | |
|---|---|
| Nome | getMonitoringKitTypes |
| URL | GET - /atticus/api/monitoringKit/getMonitoringKitTypes |
| Recupera tutti i tipi di Kit di Monitoraggio presenti nel database. | |
| Risposta | |
| STATUS 200 | <pre> application/json return List<MonitoringKitTypeDTO> [{ "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }] </pre> |

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|---|
| |] |
|--|---|

| | |
|--|---|
| Nome | getInfoMonitoringKitType |
| URL | GET - /atticus/api/monitoringKit/getInfoMonitoringKitType/{idType} |
| Recupera le informazioni sul tipo di kit di monitoraggio, in base all'id del tipo kit. | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>return Optional<MonitoringKitTypeDTO></p> <p>Esempio: /atticus/api/monitoringKit/getInfoMonitoringKitType/1</p> <pre>{ "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }</pre> |
| STATUS 500 | <p>AtticusException: MonitoringKitTypeException</p> <ul style="list-style-type: none"> Se il tipo di kit di monitoraggio non esiste viene mostrato il messaggio: "Attenzione! Il tipo del Kit di Monitoraggio non esiste" e viene mostrato il campo per il quale era stata effettuata la ricerca. |

| | |
|------|------------------|
| Nome | getMonitoringKit |
|------|------------------|

73

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| | |
|--|--|
| URL | GET - /atticus/api/monitoringKit/getMonitoringKit/{id} |
| <p>In base all'id del kit di monitoraggio, vengono recuperate le informazioni del kit: codice, descrizione, tipo, stato ed eventualmente la persona associata.</p> | |
| Risposta | |
| STATUS 200 | <pre> application/json return Optional<MonitoringKitDTO> Esempio: /atticus/api/monitoringKit/getMonitoringKit/2 { "id": 2, "code": "11:11:11", "kitTypeId": 1, "state": "A", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }, "ssnPerson": "DLEFNC96H11B519F", "personByMonitoringKit": null, "monitoringKitList": null } </pre> |

| | |
|------------|--|
| STATUS 500 | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> • Se il kit di monitoraggio non esiste: “Attenzione! Kit di Monitoraggio non esiste” • Se lo stato del kit di monitoraggio risulta in uso ma non è associato a nessun assistito: “Attenzione! Nessun’associazione Kit-Assistito: lo stato del Kit di Monitoraggio risulta in uso ma non è associato a nessun assistito” |
|------------|--|

| | |
|---|---|
| Nome | findMonitoringKitsWithPagination |
| URL | POST - /atticus/api/monitoringKit/findMonitoringKitsWithPaginatio/{allMonitoring Kit} |
| Recupera la lista paginata di tutti i kit di monitoraggio presenti nel database o la lista paginata dei kit di monitoraggio richiesti in base ai filtri inseriti. | |
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "code" }, }</pre> |

| | |
|-------------------|--|
| | <pre> "filters": { "monitoringKitCode": "11:11:11", "monitoringKitTypeId": "", } </pre> |
| <p>Risposta</p> | |
| <p>STATUS 200</p> | <pre> application/json Restituisce la lista paginata dei Kit di Monitoraggio: return Page<MonitoringKitDTO> { "content": [{ "id": 2, "code": "11:11:11", "kitTypeId": 1, "state": "A", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attivit  respiratoria - biox ancorata alla cintura." }, "ssnPerson": null, </pre> |

```
"personByMonitoringKit": null,  
"monitoringKitList": null  
}  
],  
"pageable": {  
  "sort": {  
    "sorted": true,  
    "unsorted": false,  
    "empty": false  
  },  
  "offset": 0,  
  "pageSize": 5,  
  "pageNumber": 0,  
  "unpaged": false,  
  "paged": true  
},  
"totalElements": 1,  
"totalPages": 1,  
"last": true,  
"size": 5,  
"number": 0,  
"sort": {  
  "sorted": true,  
  "unsorted": false,
```

| | |
|-------------------|--|
| | <pre> "empty": false }, "numberOfElements": 1, "first": true, "empty": false } </pre> |
| <p>STATUS 500</p> | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> • Se non è stato inserito alcun filtro: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” e la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e viene riportata la lista dei campi non validi con messaggio “Campo non valido” <p>Esempio di eccezione:</p> <pre> { "errorCode": "SEARCH_MONITORING_KIT_FILTER_ERROR", "message": "Attenzione! Campi non popolati correttamente", "fieldsError": [{ "code": "SEARCH_MONITORING_KIT_INCORRECT_CODE", "item": null, "field": "codeMonitoringKitFilter", "label": "Campo non valido" }] } </pre> |

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|--------------------------------------|
| | <pre> }] } </pre> |
|--|--------------------------------------|

| | |
|--|---|
| Nome | addMonitoringKit |
| URL | POST - /atticus/api/monitoringKit/addMonitoringKit |
| <p>Aggiunge il nuovo kit di monitoraggio, dopo aver effettuato i controlli di veridicità, obbligatorietà e unicità</p> | |
| Input | <pre> { "code": "11:00:99", "kitTypeId": 1 } </pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo del salvataggio nel database: return ResponseOperation</p> <pre> { "success": true } </pre> |
| STATUS 500 | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e viene mostrata la lista dei campi vuoti con messaggio |

79

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| | |
|--|---|
| | <p>“Campo obbligatorio” indicando il/i campo/i che ha/hanno generato l’errore</p> <ul style="list-style-type: none"> • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e viene mostrata la lista dei campi non validi con messaggio “Campo non valido” indicando il/i campo/i che ha/hanno generato l’errore • Se almeno un campo obbligatori non è popolato e almeno un campo non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e viene mostrata la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio”/“Campo non valido” <p>Esempio di eccezione:</p> <pre>{ "errorCode": "NEW_MONITORING_KIT_ERROR", "message": "Attenzione! Campi obbligatori non popolati o non popolati correttamente", "fieldsError": [{ "code": "MONITORING_KIT_EMPTY_KIT_TYPE_ID", "item": "", "field": "typeMonitoringKit", "label": "Campo obbligatorio" }, {</pre> |
|--|---|

| | |
|--|---|
| | <pre> "code": "MONITORING_KIT_INCORRECT_CODE", "item": "7686", "field": "codeMonitoringKit", "label": "Campo non valido" }] } </pre> <ul style="list-style-type: none"> • Se il codice del kit di monitoraggio inserito è già presente all'interno del sistema: "Attenzione! Kit di Monitoraggio già presente in Atticus" <p>Esempio di eccezione:</p> <pre> { "errorCode": "NEW_MONITORING_KIT_ALREADY_EXISTS", "message": "Attenzione! Kit di Monitoraggio già presente", "fieldsError": [{ "code": "MONITORING_KIT_CODE_ALREADY_EXIST", "item": "00:00:00", "field": "codeMonitoringKit", "label": " " }] } </pre> |
|--|---|

| | |
|--|---|
| | <ul style="list-style-type: none"> Se il tipo del kit di monitoraggio non è presente all'interno del sistema: "Attenzione! Il tipo del Kit di Monitoraggio non esiste" e il campo che ha generato l'errore <p>Esempio di eccezione:</p> <pre>{ "errorCode": "NOT_EXIST_KIT_TYPE_ERROR", "message": "Attenzione! Il tipo del Kit di Monitoraggio non esiste", "fieldsError": [{ "code": "NOT_EXIST_KIT_TYPE", "item": "4", "field": "typeMonitoringKit", "label": "Campo non valido" }] }</pre> |
|--|---|

| | |
|---|---|
| Nome | findMonitoringKitAssignedByFilter |
| URL | POST - /atticus/api/monitoringKit/findMonitoringKitAssignedByFilter |
| Recuperala il kit di monitoraggio in base ai filtri inseriti. | |
| Input | { "monitoringKitCode": "11:11:11", |

| | |
|------------|---|
| | <pre>"ssnPerson": "" }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json Restituisce la lista paginata dei Kit di Monitoraggio: return MonitoringKitDTO { "id": 2, "code": "11:11:11", "kitTypeId": 1, "state": "A", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - bioX ancorata alla cintura." }, "ssnPerson": null, "personByMonitoringKit": { "id": 1, "ssn": "DLEFNC96H11B519F", "idMonitoringKit": 2, "idMonitoringManager": 2, "registrationDate": 1585556393000,</pre> |

| | |
|--|---|
| | <pre> "removalDate": null, "monitoringState": "1", "idCaregiver": 4, "accessToData": 0, "idCaregiverRelationship": 1, "allowReporting": "1", "sanitaryCode": "1", "name": "Francesco", "surname": "D'Elia", "sex": "M", "birthDate": 1585748403892, "birthCity": "Campobasso", "telephoneNumber": "3657485746", "email": "francesco@delia.com", "familyDoctor": { "id": 0, "ssn": null, "idMonitoringKit": 0, "idMonitoringManager": 0, "registrationDate": null, "removalDate": null, "monitoringState": null, "idCaregiver": 0, "accessToData": 0, </pre> |
|--|---|

```
"idCaregiverRelationship": 0,
"allowReporting": null,
"sanitaryCode": null,
"name": "Pasquale",
"surname": "Sandrucci",
"sex": null,
"birthDate": null,
"birthCity": null,
"telephoneNumber": "3383647352",
"email": "pasquale@sandrucci.com",
"familyDoctor": null
}
},
"monitoringKitList": [
{
  "id": 3,
  "code": "22:22:22",
  "kitTypeId": 1,
  "state": "D",
  "monitoringKitTypeByKitTypeId": {
    "id": 1,
    "name": "Smart t-shirt",
    "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4
elettrodi per attività respiratoria - biox ancorata alla cintura."
  }
}
```

| | |
|-------------------|--|
| | <pre> }, "personByKitCode": null }, { "id": 4, "code": "33:33:33", "kitTypeId": 1, "state": "D", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }, "personByKitCode": null }] } </pre> |
| <p>STATUS 500</p> | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> Se non è stato inserito alcun filtro: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” e la lista dei campi vuoti con messaggio “Campo obbligatorio” |

- Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido”

Esempio di eccezione:

```
{
  "errorCode": "SEARCH_MONITORING_KIT_FILTER_ERROR",
  "message": "Attenzione! Selezionare almeno un campo per effettuare la
ricerca",
  "fieldsError": [
    {
      "code": "SEARCH_MONITORING_KIT_EMPTY_CODE",
      "item": null,
      "field": "codeMonitoringKitFilter",
      "label": "Campo obbligatorio"
    },
    {
      "code": "SEARCH_MONITORING_KIT_EMPTY_SSN_PERSON",
      "item": null,
      "field": "ssnPersonAssignedMonitoringKitFilter",
      "label": "Campo obbligatorio"
    }
  ]
}
```

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|--|
| | <ul style="list-style-type: none"> • Se non viene recuperato nessun kit di monitoraggio con i criteri di ricerca inseriti: "Attenzione! Nessun Kit di Monitoraggio trovato corrispondente ai criteri di ricerca inseriti" • Se nel sistema non è presente nessun kit di monitoraggio disponibile: "Attenzione! Non ci sono Kit di Monitoraggio disponibili al momento" |
|--|--|

| | |
|---|--|
| Nome | replaceMonitoringKit |
| URL | POST - /atticus/api/monitoringKit/replaceMonitoringKit |
| Recupera il kit di monitoraggio in base ai filtri inseriti. | |
| Input | <pre>{ "personId": 1, "replacedMonitoringKitId": 2, "replacedMonitoringKitState": "D", "newMonitoringKitAssignedId": 1 }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json Ritorna il successo del salvataggio nel database: return ResponseOperation { "success": true }</pre> |

| | |
|------------|--|
| STATUS 500 | <p>AtticusException: PersonException</p> <ul style="list-style-type: none"> • Se non viene trovato nessun assistito: “Attenzione! Assistito non presente in Atticus” • AtticusException: KitAssignedException • Se non viene trovato alcun’associazione kit-assistito: “Attenzione! Assegnazione Kit-Assistito non trovata” |
|------------|--|

| | |
|---|--|
| Nome | updateMonitoringKit |
| URL | PUT -/atticus/api/monitoringKit/updateMonitoringKit |
| <p>In caso di superamento dei controlli di veridicità, obbligatorietà e univocità, il sistema aggiorna il kit di monitoraggio riportando le modifiche effettuate.</p> | |
| Input | <pre>{ "id": 1, "code": "00:00:01", "kitTypeId": 1, "state": "E", "monitoringKitTypeByKitTypeId": { "id": 1, "name": "Smart t-shirt", "description": "T-shirt manica lunga con 4 elettrodi per ECG e 4 elettrodi per attività respiratoria - biox ancorata alla cintura." }, }</pre> |

| | |
|------------|--|
| | <pre>"ssnPerson": null }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json</pre> <p>Ritorna il successo del salvataggio nel database: return ResponseOperation</p> <pre>{ "success": true }</pre> |
| STATUS 500 | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e viene riportata la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e viene riportata la lista dei campi non validi con messaggio “Campo non valido” • Se almeno un campo obbligatorio non è popolato e almeno un campo non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e viene riportata la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio” / “Campo non valido” <p>Esempio di eccezione:</p> <pre>{</pre> |

```
"errorCode": "NEW_MONITORING_KIT_ERROR",
"message": "Attenzione! Campi obbligatori non popolati o non popolati
correttamente",
"fieldsError": [
  {
    "code": "MONITORING_KIT_EMPTY_KIT_TYPE_ID",
    "item": "",
    "field": "typeMonitoringKit",
    "label": "Campo obbligatorio"
  },
  {
    "code": "MONITORING_KIT_INCORRECT_CODE",
    "item": "7686",
    "field": "codeMonitoringKit",
    "label": "Campo non valido"
  }
]
}
```

- Se il codice del kit di monitoraggio inserito è già presente all'interno del sistema: "Attenzione! Kit di Monitoraggio già presente in Atticus"

Esempio di eccezione:

```
{
"errorCode": "NEW_MONITORING_KIT_ALREADY_EXISTS",
```

```
"message": "Attenzione! Kit di Monitoraggio già presente",
"fieldsError": [
  {
    "code": "MONITORING_KIT_CODE_ALREADY_EXIST",
    "item": "00:00:00",
    "field": "codeMonitoringKit",
    "label": " "
  }
]
```

- Se il tipo del kit di monitoraggio non è presente all'interno del sistema:
"Attenzione! Il tipo del Kit di Monitoraggio non esiste"

Esempio di eccezione:

```
{
  "errorCode": "NOT_EXIST_KIT_TYPE_ERROR",
  "message": "Attenzione! Kit di Monitoraggio non esistente",
  "fieldsError": [
    {
      "code": "MONITORING_KIT_CODE_ALREADY_EXIST",
      "item": "00:00:00",
      "field": "codeMonitoringKit",
      "label": " "
    }
  ]
}
```

| | |
|--|---|
| | <pre>] } </pre> <ul style="list-style-type: none"> Se il codice del kit di monitoraggio inserito è già presente all'interno del sistema: "Attenzione! Kit di Monitoraggio già presente in Atticus" <p>Esempio di eccezione:</p> <pre> { "errorCode": "NEW_MONITORING_KIT_ALREADY_EXISTS", "message": "Attenzione! Kit di Monitoraggio già presente", "fieldsError": [{ "code": "MONITORING_KIT_CODE_ALREADY_EXIST", "item": "00:00:00", "field": "codeMonitoringKit", "label": " " }] } </pre> <ul style="list-style-type: none"> Se il tipo del kit di monitoraggio non è presente all'interno del sistema: "Attenzione! Il tipo del Kit di Monitoraggio non esiste" <p>Esempio di eccezione:</p> <pre> { "errorCode": "NOT_EXIST_KIT_TYPE_ERROR", "message": "Attenzione! Kit di Monitoraggio non esistente", </pre> |
|--|---|

| | |
|--|---|
| | <pre> "fieldsError": [{ "code": "MONITORING_KIT_CODE_ALREADY_EXIST", "item": "00:00:00", "field": "codeMonitoringKit", "label": " " }] </pre> |
|--|---|

| | |
|---|---|
| Nome | deleteMonitoringKit |
| URL | DELETE - /atticus/api/monitoringKit/deleteMonitoringKit/{id} |
| Elimina nel database il kit di monitoraggio modificato, sulla base dell'id kit. | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo dell'eliminazione nel database: return ResponseOperation</p> <p>Esempio: /atticus/api/monitoringKit/deleteMonitoringKit/1</p> <pre> { "success": true } </pre> |
| STATUS 500 | AtticusException: MonitoringKitException |

| | |
|--|--|
| | <ul style="list-style-type: none"> Se il kit di monitoraggio non esiste nel sistema: “Attenzione! Kit di Monitoraggio non esiste” |
|--|--|

3.3 Servizi arruolamento

| | |
|--|--|
| Nome | getPersonToEnrollment |
| URL | GET - /atticus/api/person/getPersonToEnrollment/{ssn}&{sanitaryCode} |
| Recupera le informazioni dell’assistito sulla base del codice fiscale e codice sanitario | |
| Risposta | |
| STATUS 200 | <p>application/json – Restituisce l’Assistito</p> <p>return List<PersonDTO></p> <p>Esempio:</p> <p>http://localhost:9090/atticus/api/person/getPersonToEnrollment/2&</p> <pre>[{ "id": 1, "ssn": "DLEMRT11H41B519G", "nameDB": null, "surnameDB": null, "healthCode": null, "idMonitoringKit": 0, "idMonitoringManager": 0, "registrationDate": null,</pre> |

| | |
|--|---|
| | <pre>"removalDate": null, "monitoringState": null, "idCaregiver": 0, "accessToData": 0, "idCaregiverRelationship": 0, "allowReporting": null, "sanitaryCode": "2", "name": "Martina", "surname": "D'Elia", "sex": "F", "birthDate": "2020-04-14T13:47:53.315+0000", "birthCity": "Campobasso", "telephoneNumber": "3768564735", "email": "martina@delia.com", "familyDoctor": { "id": 0, "ssn": null, "nameDB": null, "surnameDB": null, "healthCode": null, "idMonitoringKit": 0, "idMonitoringManager": 0, "registrationDate": null, "removalDate": null,</pre> |
|--|---|

| | |
|------------|--|
| | <pre> "monitoringState": null, "idCaregiver": 0, "accessToData": 0, "idCaregiverRelationship": 0, "allowReporting": null, "sanitaryCode": null, "name": "Nicola", "surname": "Pasquarelli", "sex": null, "birthDate": null, "birthCity": null, "telephoneNumber": "3454657343", "email": "nicola@sandrucci.com", "familyDoctor": null } }] </pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> • Se i filtri inseriti non sono validi :<i>“Filtri inseriti non validi”</i>.. • Se l’assistito è stato già censito :<i>“Assisito gia' presente in Atticus”</i>. |

| | |
|------|--------|
| Nome | enroll |
|------|--------|

| | |
|--|--|
| URL | POST - atticus/api/person/enrollment/enroll |
| <p>Dopo aver eseguito, ove richiesto, i controlli di univocità, obbligatorietà e validità e compilati i passi sequenziali richiesti dal sistema, viene effettuato l'arruolamento dell'assistito.</p> | |
| Input | <pre>{ "person":{ "id":1, "ssn":"DLEMRT11H41B519G", "nameDB":"Martina", "surnameDB":"D'Elia", "healthCode":"2", "idMonitoringKit":42, "idMonitoringManager":2, "registrationDate":null, "removalDate":null, "monitoringState":null, "idCaregiver":3, "accessToData":0, "idCaregiverRelationship":1, "allowReporting":"0", "sanitaryCode":"2", "name":"Martina", "surname":"D'Elia", "sex":"F",</pre> |

| | |
|--|--|
| | <pre> "birthDate":"2020-04-14T15:15:29.166Z", "birthCity":"Campobasso", "telephoneNumber":"3768564735", "email":"martina@delia.com", "familyDoctor":{ "name":"Nicola", "surname":"Pasquarelli", "telephoneNumber":"3454657343", "email":"nicola@sandrucchi.com", } }, "doctorToPerson":[{"idUser":4}], "assignedKit":{ "idKit":42 }, "checkUp":{ "weight":80, "height":"180", "dbpOrtostatic":"5", "dbpSupine":"4", "dyuresis":"21", "glycemia":"10", </pre> |
|--|--|

| | |
|--|---|
| | <pre> "idPerson":1, "sbpOrtostatic":"11", "sbpSupine":"51", "spO2":"99" }, "checkUpToPathology":[{ "idPathology":1}], "alarmThresholds":{ "idKit":42, "personSsn":"", "heartAlerts":[{"type":"tachyarrhythmia","level":3,"lowerBound":110,"upperBound":119}, {"type":"tachyarrhythmia","level":2,"lowerBound":120,"upperBound":130}, {"type":"tachyarrhythmia","level":1,"lowerBound":999,"upperBound":0}, {"type":"bradyarrhythmia","level":3,"lowerBound":46,"upperBound":55},] } </pre> |
|--|---|

| | |
|--|--|
| | <pre> {"type":"bradyarrhythmia","level":2,"lowerBound":41,"upperBound":45}, {"type":"bradyarrhythmia","level":1,"lowerBound":0,"upperBound":50}],"breathAlerts":[{"type":"tachypnoea","level":2,"lowerBound":21,"upperBound":25}, {"type":"tachypnoea","level":1,"lowerBound":0,"upperBound":25}, {"type":"bradypnea","level":2,"lowerBound":10,"upperBound":15}, {"type":"bradypnea","level":1,"lowerBound":0,"upperBound":10}], "lastUpdateTime":0}, "instrumentalTests":["bloodTests":[{"editDisabled":true,"inputError":false,"idType":1,"value":"20","examType":{"id":1,"name":"Creatinina","measureUnit":"mg/dl"},"date":"2020-04-23T22:00:00.000Z","idPerson":1,"idUser":1}], </pre> |
|--|--|

| | |
|-------------------|--|
| | <pre> "therapies":[{"dose":"Dose","drug":{"id":1,"name":"tachipirina"},"idDurationType":1,"mode":"Ass","idDrug":1,"value":"2","time":{"id":1,"name":"Settimana"},"startDate":"2020-04-13T22:00:00.000Z","endDate":"2020-04-28T15:19:49.425Z","idPerson":1,"idUser":1}], "appointment":{ "date":"2020-04-13T22:00:00.000Z", "note":"Note", "idPerson":1, "idUser":1 } } </pre> |
| <p>Risposta</p> | |
| <p>STATUS 200</p> | <pre> application/json – Restituisce la lista paginata dei medicinali return ResponseOperation { "success": true } </pre> |
| <p>STATUS 500</p> | <p>AtticusException:</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori è vuoto. |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Se uno dei campi non è popolato correttamente. |
|--|--|

3.4 Servizi gestione assistiti

| | |
|---|---|
| Nome | getPerson |
| URL | GET - /atticus/api/person/getPerson/{idPerson} |
| Recupera le informazioni dell'Assistito, sulla base dell'id dell'Assistito. | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce l'Assistito: return PersonDTO</p> <p>Esempio: /atticus/api/person/getPerson/1</p> <pre>{ "id": 1, "ssn": "DLEFNC96H11B519F", "name": "Francesco", "surname": "D'Elia", "healthCode": "1", "idMonitoringKit": 1, "idMonitoringManager": 2, "registrationDate": "2020-04-20T07:54:27.000+0000", "removalDate": null, "monitoringState": "1", "idCaregiver": 3,</pre> |

| | |
|------------|--|
| | <pre> "accessToData": 1, "idCaregiverRelationship": 3, "allowReporting": "1", "externalPerson": { "healthCode": "1", "ssn": "DLEFNC96H11B519F", "name": "Francesco", "surname": "D'Elia", "sex": "M", "birthDate": "2020-04-28T13:41:39.035+0000", "birthCity": "Campobasso", "telephoneNumber": "3657485746", "email": "francesco@delia.com", "familyDoctor": { "name": "Pasquale", "surname": "Sandrucci", "telephoneNumber": "3383647352", "email": "pasquale@sandrucci.com" } } } </pre> |
| STATUS 500 | AtticusException: PersonException |

| | |
|--|---|
| | Se l'assistito non è presente all'interno de sistema: "Attenzione! L'Assistito non esiste in Atticus" |
|--|---|

| | |
|---|--|
| Nome | findPersonsWithPagination |
| URL | POST - /atticus/api/user/findPersonsWithPagination |
| Recuperala lista paginata degli assistiti presenti nel database in base ai filtri inseriti. | |
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "ssn" }, "filters": { "surname": "d'elia", "name": "", "ssn": "", "healthCode": "" } }</pre> |
| Risposta | |
| STATUS 200 | application/json |

Restituisce la lista paginata degli Utenti: return Page<UserDTO>

```
{
  "content": [
    {
      "id": 1,
      "ssn": "DLEFNC96H11B519F",
      "name": "Francesco",
      "surname": "D'Elia",
      "healthCode": "1",
      "idMonitoringKit": 1,
      "idMonitoringManager": 2,
      "registrationDate": "2020-04-20T07:54:27.000+0000",
      "removalDate": null,
      "monitoringState": "1",
      "idCaregiver": 3,
      "accessToData": 1,
      "idCaregiverRelationship": 3,
      "allowReporting": "1",
      "externalPerson": null
    },
    {
      "id": 2,
      "ssn": "DLEMRT11H41B519G",
      "name": "Martina",
```

```
"surname": "D'Elia",
"healthCode": "2",
"idMonitoringKit": 3,
"idMonitoringManager": 2,
"registrationDate": "2020-04-22T14:49:45.000+0000",
"removalDate": null,
"monitoringState": "1",
"idCaregiver": 3,
"accessToData": 1,
"idCaregiverRelationship": 4,
"allowReporting": "1",
"externalPerson": null
}
],
"pageable": {
  "sort": {
    "sorted": true,
    "unsorted": false,
    "empty": false
  },
  "offset": 0,
  "pageSize": 5,
  "pageNumber": 0,
  "unpaged": false,
```

| | |
|-------------------|---|
| | <pre> "paged": true }, "totalPages": 1, "totalElements": 2, "last": true, "size": 5, "number": 0, "sort": { "sorted": true, "unsorted": false, "empty": false }, "numberOfElements": 2, "first": true, "empty": false } </pre> |
| <p>STATUS 500</p> | <p>AtticusException: PersonException</p> <ul style="list-style-type: none"> • Se non è stato inserito alcun filtro: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” e la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore |

| | |
|--|--|
| | <p>Esempio di eccezione:</p> <pre>{ "errorCode": "SEARCH_PERSON_FILTER_ERROR", "message": "Attenzione! Campi non popolati correttamente", "fieldsError": [{ "code": "SEARCH_PERSON_INCORRECT_SURNAME", "item": "616", "field": "surnamePersonFilter", "label": "Campo non valido" }] }</pre> |
|--|--|

| | |
|--|--|
| Nome | changeMonitoringState |
| URL | POST - /atticus/api/person/changeMonitoringState |
| Cambia lo stato del monitoraggio dell'assistito da attivo a non attivo o viceversa | |
| Input | <pre>{ "idPerson": 1, "active": true, "reason": "Non occorre piu' il monitoraggio" }</pre> |

| Risposta | |
|------------|--|
| STATUS 200 | application/json Ritorna il successo del salvataggio nel database: return ResponseOperation <pre>{ "success": true }</pre> |
| STATUS 500 | Atticus Exception: PersonException <ul style="list-style-type: none"> • Se l'assistito non è presente all'interno del sistema: "Attenzione! Assistito non presente" • Se lo stato del monitoraggio dell'assistito è attivo: "Attenzione! Lo stato del monitoraggio dell'Assistito risulta attivo" |

| Nome | deletePerson |
|--|--|
| URL | DELETE - /atticus/api/person/deletePerson/{idPerson} |
| Effettua l'eliminazione logica nel database l'assistito, sulla base dell'id. | |
| Input | <pre>{ "idPerson": 1, "active": true, "reason": "Non occorre piu' il monitoraggio" }</pre> |
| Risposta | |

| | |
|-------------------|---|
| <p>STATUS 200</p> | <p>application/json</p> <p>Ritorna il successo dell'eliminazione nel database: return ResponseOperation</p> <p>Esempio: /atticus/api/person/deletePerson/1</p> <pre>{ "reason": "E' deceduto" }</pre> |
| <p>STATUS 500</p> | <p>AtticusException: PersonException</p> <ul style="list-style-type: none"> • Se l'assistito non è presente all'interno del sistema: "Attenzione! Assistito non presente" • Se lo stato del monitoraggio dell'assistito è attivo: "Attenzione! Lo stato del monitoraggio dell'Assistito risulta attivo" <pre>{ "errorCode": "DELETE_PERSON_ENABLED_MONITORING_STATE", "message": "Attenzione! Lo stato del monitoraggio dell'Assistito risulta attivo", "fieldsError": null }</pre> <ul style="list-style-type: none"> • Se non si riesce a risalire all'utente loggato che sta tentando di eliminare l'assistito: "Attenzione! Impossibile eliminare l'assistito" |

| | |
|-------------|---|
| <p>Nome</p> | <p>findPersonAssociationsWithPagination</p> |
|-------------|---|

| | |
|---|---|
| URL | POST - /atticus/api/person/findPersonAssociationsWithPagination/{idPerson} |
| Recupera la lista paginata degli utenti associati all'assistito per il quale si desidera conoscere le associazioni. | |
| Input | { "pageNumber": 0, "pageSize": 5, "sortedBy": "surname" } |
| Risposta | |
| STATUS 200 | application/json Restituisce la lista paginata degli Utenti: return Page<UserAssociationsDTO> { "content": [{ "id": 4, "name": "Federica", "surname": "Genovese", "telephoneNumber": "3331122331", "email": "federica@genovese.it", "idRole": 4, |

| | |
|--|---|
| | <pre>"accessToData": 0, "idCaregiverRelationship": 0, "editDisabled": true, "inputErrorEmail": false, "inputErrorTelephoneNumber": false }, { "id": 3, "name": "Giuseppe", "surname": "Battista", "telephoneNumber": "3331122378", "email": "giuseppe@battista.it", "idRole": 2, "accessToData": 1, "idCaregiverRelationship": 3, "editDisabled": true, "inputErrorEmail": false, "inputErrorTelephoneNumber": false }, { "id": 0, "name": "Pasquale", "surname": "Sandrucci", "telephoneNumber": "3383647352",</pre> |
|--|---|

```
"email": "pasquale@sandrucci.com",
"roleId": 1,
"accessToData": 0,
"idCaregiverRelationship": 0,
"editDisabled": true,
"inputErrorEmail": false,
"inputErrorTelephoneNumber": false
}
],
"pageable": {
  "sort": {
    "sorted": true,
    "unsorted": false,
    "empty": false
  },
  "offset": 0,
  "pageSize": 5,
  "pageNumber": 0,
  "unpaged": false,
  "paged": true
},
"totalPages": 1,
"totalElements": 3,
"last": true,
```

| | |
|------------|--|
| | <pre>"size": 5, "number": 0, "sort": { "sorted": true, "unsorted": false, "empty": false }, "numberOfElements": 3, "first": true, "empty": false }</pre> |
| STATUS 500 | <p>AtticusException: PersonException</p> <p>Se l'assistito non è presente: "Attenzione! L'Assistito non esiste in Atticus"</p> |

| | |
|-------|--|
| Nome | updateAdministrativeData |
| URL | POST - /atticus/api/person/updateAdministrativeData |
| | Dopo aver effettuato i controlli di veridicità, obbligatorietà e unicità modifica informazioni inerenti gli utenti associati all'assistito oppure aggiunge nuovi utenti associati all'assistito. |
| Input | <pre>{ "person": { "id": 1, "ssn": "DLEFNC96H11B519F",</pre> |

```
"name": "Francesco",
"surname": "D'Elia",
"healthCode": "1",
"idMonitoringKit": 1,
"idMonitoringManager": 2,
"registrationDate": "2020-04-20T07:54:27.000+0000",
"removalDate": null,
"monitoringState": "0",
"idCaregiver": 3,
"accessToData": 1,
"idCaregiverRelationship": 3,
"allowReporting": "1"
},
"idCaregiverToRemove": 3,
"caregiverToAdd": {
  "id": 9,
  "idRole": 2,
  "idResidenceCity": 5675,
  "ssn": "DSCPNI65R62B519S",
  "surname": "de Socio",
  "name": "Pina",
  "sex": "F",
  "birthDate": "1965-10-21T23:00:00.000+0000",
  "idBirthPlace": 5675,
```

```
"birthState": "IT",
"abroadBirthPlace": null,
"residenceAddress": "Via Roma",
"residencePc": "12345",
"cellNumber": "3332547885",
"email": "pina@desocio.it",
"suspended": "0",
"doctorCode": null,
"cityByIdResidenceCity": {
  "id": 5675,
  "name": "Campobasso",
  "province": "Campobasso",
  "provinceAbbreviation": "CB"
},
"cityByIdBirthPlace": {
  "id": 5675,
  "name": "Campobasso",
  "province": "Campobasso",
  "provinceAbbreviation": "CB"
},
"caragiverRelationship": {
  "id": 1,
  "name": "Coniuge"
}
```

| | |
|------------|---|
| | <pre> }, "caregiverUpdate": null, "idSpecializedDoctorsToRemove": [], "idSpecializedDoctorsToAdd": [], "idMonitoringManagerToRemove": null, "idMonitoringManagerToAdd": null } </pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo del salvataggio nel database: return ResponseOperation</p> <pre> { "success": true } </pre> |
| STATUS 500 | <p>AtticusException: PersonException</p> <ul style="list-style-type: none"> • Se l'assistito non è presente all'interno de sistema: "Attenzione! Assistito non presente" • Se il caregiver che si vuole associare all'assistito non è presente all'interno de sistema: "Attenzione! Il Caregiver che si desidera associare all'Assistito non esiste in Atticus" • Se viene eliminato il caregiver attualmente associato e non viene associato un nuovo caregiver all'assistito: "Attenzione! Bisogna associare un Caregiver all'Assistito" |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Se il caregiver che si vuole modificare non è presente all'interno de sistema: "Attenzione! Il Caregiver che si desidera aggiornare all'Assistito non esiste in Atticus" • Se vengono eliminati tutti i medici specialisti associati all'assistito e non ne viene associato nemmeno un medico specialista: "Attenzione Bisogna associare almeno un Medico Specialista all'Assistito" |
|--|---|

| | |
|--|--|
| Nome | getCheckUp |
| URL | GET - /atticus/api/checkUp/getCheckUp/{checkUpId} |
| Recupera le informazioni di un particolare checkup | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce il check-up: return CheckUpDTO</p> <pre>{ "date": "2020-04-29T22=00=00.000+0000", "dbpOrtostatic": "67", "dbpSupine": "123", "dyuresis": "null", "glycemia": "null", "height": 180, "id": 175, "idAppointment": "null",</pre> |

| | |
|------------|---|
| | <pre> "idPerson": 154, "idUser": 2, "pathologies": [{ "id": 1, "name": "Tachicardia" }], "sbpOrtostatic": "78", "sbpSupine": "145", "spO2": 99, "user": "null", "weight": 80 } </pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> “Attenzione! Impossibile recuperare il check-up” Check-up non trovato |

| | |
|-----------------------------------|---|
| Nome | getPathologies |
| URL | GET - /atticus/api/pathology/getPathologies |
| Recupera le patologie disponibili | |
| Risposta | |

| | |
|------------|--|
| STATUS 200 | <p>application/json</p> <p>Restituisce le patologie: return List<PathologyDTO></p> <p>Esempio: /atticus/api/pathology/getPathologies</p> <pre>[{ "id": 1, "name": "Raffreddore" }]</pre> |
| STATUS 500 | AtticusException |

| | |
|---|--|
| Nome | addCheckUp |
| URL | POST - /atticus/api/checkUp/save |
| Aggiunge un nuovo follow-up all'assistito selezionato | |
| Input | <pre>{ "checkUp": { "height": "80", "idPerson": "154", "weight": "80" }, "checkUpToPathology": [</pre> |

| | |
|------------|--|
| | <pre>{ "idPathology": "2" }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce il successo del salvataggio: return boolean</p> <p>true</p> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> • Se non vengono correttamente popolati i campi: “Attenzione! Impossibile salvare il check-up” Errore nel salvataggio del check-up • Se non esiste alcun appuntamento associato al check up: “Attenzione! Appuntamento non trovato” • Se i campi sono vuoti e non popolati correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente”. • Se esiste almeno un campo obbligatorio non popolato: “Attenzione! Campi obbligatori non popolati” • Se esiste almeno un campo non popolato correttamente: “Attenzione! Campi non popolati correttamente” |

| | |
|------|------------|
| Nome | addCheckUp |
|------|------------|

| | |
|---|---|
| URL | POST - /atticus/api/checkUp/getCheckUpsToHistory |
| Recupera tutti i follow-up di un assistito specificando eventuali date. | |
| Input | <pre>{ "filters": { "dateFrom": "2020-04-12T22=00=00.000Z", "dateTo": "2020-04-29T22=00=00.000Z", "personId": 154 }, "paginationAndSortingDetails": [{ "desc": "false", "pageNumber": 0, "pageSize": 5, "sortedBy": "date" }] }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json Restituisce la paginazione dei check up: return Page<CheckUpDTO> {</pre> |

```
"content": [  
  {  
    "date": "2020-04-29T22=00=00.000+0000",  
    "dbpOrtostatic": "null",  
    "dbpSupine": "null",  
    "dyuresis": "null",  
    "glycemia": "null",  
    "height": 180,  
    "id": 175,  
    "idAppointment": "null",  
    "idPerson": 154,  
    "idUser": 2,  
    "pathologies": "null",  
    "sbpOrtostatic": "null",  
    "sbpSupine": "null",  
    "sp02": 99,  
    "user": [  
      {  
        "abroadBirthPlace": "null",  
        "accessToData": "false",  
        "birthDate": "1972-07-18T23=00=00.000+0000",  
        "birthState": "IT",  
        "caragiverRelationship": "null",  
        "cellNumber": "3333434000",
```

| | |
|--|--|
| | <pre>"cityByIdBirthPlace": [{ "id": 5670, "name": "AcquavivaCollecroce", "province": "Campobasso", "provinceAbbreviation": "CB" }], "cityByIdResidenceCity": [{ "id": 5670, "name": "AcquavivaCollecroce", "province": "Campobasso", "provinceAbbreviation": "CB" }], "doctorCode": 1111, "email": "federica@rossi.it", "hasAssistedAssociated": "false", "id": 2, "idBirthPlace": 5670, "idResidenceCity": 5670, "idRole": 5, "name": "Federica",</pre> |
|--|--|

```
"residenceAddress": "viaXXIVMaggio",  
"residencePc": "86100",  
"sex": "F",  
"ssn": "RSSFRC72L53B519U",  
"surname": "Rossi",  
"suspended": 0,  
"weight": 80  
}  
],  
{  
"date": "2020-04-29T22=00=00.000+0000",  
"dbpOrtostatic": "null",  
"dbpSupine": "null",  
"dyuresis": "null",  
"glycemia": "null",  
"height": 180,  
"id": 175,  
"idAppointment": "null",  
"idPerson": 154,  
"idUser": 2,  
"pathologies": "null",  
"sbpOrtostatic": "null",  
"sbpSupine": "null",
```

```
"sp02": 99,
"user": [
  {
    "abroadBirthPlace": "null",
    "accessToData": "false",
    "birthDate": "1972-07-18T23=00=00.000+0000",
    "birthState": "IT",
    "caragiverRelationship": "null",
    "cellNumber": "3333434000",
    "cityByIdBirthPlace": [
      {
        "id": 5670,
        "name": "AcquavivaCollecroce",
        "province": "Campobasso",
        "provinceAbbreviation": "CB"
      }
    ],
    "cityByIdResidenceCity": [
      {
        "id": 5670,
        "name": "AcquavivaCollecroce",
        "province": "Campobasso",
        "provinceAbbreviation": "CB"
      }
    ]
  }
]
```

```
],  
  "doctorCode": 1111,  
  "email": "federica@rossi.it",  
  "hasAssistedAssociated": "false",  
  "id": 2,  
  "idBirthPlace": 5670,  
  "idResidenceCity": 5670,  
  "idRole": 5,  
  "name": "Federica",  
  "residenceAddress": "viaXXIVMaggio",  
  "residencePc": "86100",  
  "sex": "F",  
  "ssn": "RSSFRC72L53B519U",  
  "surname": "Rossi",  
  "suspended": 0,  
  "weight": 80  
}  
]  
}  
],  
  "empty": "false",  
  "first": "true",  
  "last": "false",  
  "number": 0,
```

| | |
|------------|---|
| | <pre>"numberOfElements": 5, "size": 5, "totalElements": 6, "totalPages": 2 }</pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> • Se non vengono popolati i campi di ricerca: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” • Se i campi di ricerca non sono popolati correttamente: “Attenzione! Campi non popolati correttamente” |

| | |
|--|--|
| Nome | getAlarmThresholds |
| URL | GET - /atticus/public/alarmThresholds/getAlarmThresholds/{kitId} |
| Recupera le soglie d’allarme di un assistito | |
| Risposta | |
| STATUS 200 | <pre>application/json Restituisce le soglie d’allarme dell’assistito: return AlarmThresholds { "breathAlerts": [{ "level": 2,</pre> |

```

"lowerBound": 21,
"type": "tachypnoea",
"upperBound": "25"
},
{
"level": 1,
"lowerBound": 0,
"type": "tachypnoea",
"upperBound": 25
},
{
"level": 2,
"lowerBound": 10,
"type": "bradypnea",
"upperBound": 15
},
{
"level": 1,
"lowerBound": 0,
"type": "bradypnea",
"upperBound": 10
}
],
"heartAlerts": [

```

```
{
  "level": 3,
  "lowerBound": 110,
  "type": "tachyarrhythmia",
  "upperBound": 128
},
{
  "level": 2,
  "lowerBound": 129,
  "type": "tachyarrhythmia",
  "upperBound": 309
},
{
  "level": 1,
  "lowerBound": 310,
  "type": "tachyarrhythmia",
  "upperBound": 0
},
{
  "level": 3,
  "lowerBound": 46,
  "type": "bradyarrhythmia",
  "upperBound": 55
},
```

| | |
|------------|--|
| | <pre>{ "level": 2, "lowerBound": 41, "type": "bradyarrhythmia", "upperBound": 45 }, { "level": 1, "lowerBound": 0, "type": "bradyarrhythmia", "upperBound": 50 }], "kitId": 41, "lastUpdateTime": 1588064851493, "personSsn": "", "_id": "5ea7edb559199f36a8792cb9" }</pre> |
| STATUS 500 | AtticusException: |

| | |
|------|--|
| Nome | addExam |
| URL | POST - /atticus/api/instrumentalTest/addExam |

| Aggiunge un nuovo esame strumentale | |
|-------------------------------------|---|
| Input | <pre>{ "file":{ "_files":{}, "delimiter":",", "_fileNames":"grafico-normale-di-battito-cardiaco-ecg-71108214.jpg" }, "examType": { "id":1, "name":"Ecg" }, "date":"2020-04-15T02:00:00.000Z", "idType":1, "fileUrl":"/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 15-04-2020 04-00-00.jpg", "idPerson":140 }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce l'esame strumentale salvato: return InstrumentalTestDTO</p> <pre>{</pre> |

| | |
|------------|--|
| | <pre> "id": 71, "date": "2020-04-15T02:00:00.000+0000", "fileUrl": "/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 15-04-2020 04-00-00.jpg" "idType": 1 "report": null "idUser": 2 "idPerson": 140 "user": null "examType": null } </pre> |
| STATUS 500 | <ul style="list-style-type: none"> AtticusException: Se uno dei campi non è popolato correttamente |

| | |
|---|---|
| Nome | getInstrumentalTestsToComparing |
| URL | POST - /atticus/api/instrumentalTest/getInstrumentalTestsToComparing |
| Recupera la lista degli esami strumentali da confrontare. | |
| Input | <pre> { "idType":1, "personId":"140" } </pre> |
| Risposta | |

| | |
|------------|---|
| STATUS 200 | <p>application/json</p> <p>Restituisce gli esami strumentali: return List<InstrumentalTestDTO></p> <pre> [[{"id": "45", "date": "2020-04-24T02=00=00.000+0000", "fileUrl": "/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 24-04-2020 04-00-00.jpg", "idType": "1", "report": "null", "idUser": "2", "idPerson": "140", "user": "null", "examType": "null"} { "id": "71", "date": "2020-04-15T02=00=00.000+0000", "fileUrl": "/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 15-04-2020 04-00-00.jpg", "idType": "1", "report": "null", "idUser": "2", "idPerson": "140", "user": "null", "examType": "null" }]] </pre> |
|------------|---|

| | |
|------------|---|
| | <pre>}},] }</pre> |
| STATUS 500 | <ul style="list-style-type: none"> AtticusException: Se uno dei campi non è popolato correttamente |

| | |
|--|--|
| Nome | getImagesToComparing |
| URL | POST - /atticus/api/images/getImagesToComparing |
| Recupera le immagini dei tipi di esame da confrontare. | |
| Input | <pre>{ "idType":1, "personId":"140" }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json Restituisce le immagini da comparare: return List<ImageDTO> [[image: "iVBORw0KGgoAAAANSUhEUgAAAs8AAAGvCAMAAACTsMhCAAAABG", image: "iVBORw0KGgoAAAANSUhEUgAAAs8AAAGvCAMAAACTsMhCAAAABG"</pre> |

| | |
|------------|--|
| | <pre>]} </pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> Se non è possibile recuperare l'immagine: "Impossibile recuperare l'immagine" |

| | |
|--|---|
| Nome | deleteExam |
| URL | DELETE - /atticus/api/instrumentalTest/deleteExam/{examId} |
| Elimina l'esame strumentale precedentemente inserito | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Ritorna il successo dell'eliminazione nel database: return boolean</p> <p>Esempio: /atticus/api/instrumentalTest/deleteExam/1</p> <p>true</p> |
| STATUS 500 | <p>AtticusException: MonitoringKitException</p> <ul style="list-style-type: none"> Errore durante l'eliminazione: "Attenzione! Impossibile eliminare l'esame strumentale". |

| | |
|------|--|
| Nome | savelmage |
| URL | POST - /atticus/api/images/savelmage/{ssn} |

| Carica l'immagine dell'esame inserito | |
|---------------------------------------|---|
| Input | <pre>{ "image": byte[] }</pre> |
| Risposta | |
| STATUS 200 | application/json Restituisce il successo nel salvataggio dell'immagine: return boolean true |
| STATUS 500 | AtticusException: <ul style="list-style-type: none"> Se dovessero presentarsi problemi durante il salvataggio di un file: "false". |

| Nome | getImage |
|---------------------------------|--|
| URL | POST - /atticus/api/images/getImage |
| Recupera l'immagine di un esame | |
| Input | <pre>{ /ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 20-04-2020 05-00-00.png }</pre> |
| Risposta | |
| STATUS 200 | application/json |

| | |
|------------|---|
| | <p>Restituisce l'immagine: return <ImageDTO></p> <pre>{ image: "iVBORw0KGgoAAAANSUhEUgAAAs8AAAGvCAMAAACTsMhCAAAABG" }</pre> |
| STATUS 500 | RuntimeException |

| | |
|---|--|
| Nome | getBloodTestTypes |
| URL | GET - /atticus/api/bloodTestType/getBloodTestTypes |
| Recupera gli esami ematochimici presenti in Atticus | |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce i tipi degli esami ematochimici: return List<BloodTestTypeDTO></p> <pre>[{ "id": 1, "name": "Creatinina", "measureUnit": "mg/dl" }, { "id": 3,</pre> |

| | |
|------------|---|
| | <pre> "name": "Potassio", "measureUnit": "mEq/l" }, { "id": 2, "name": "Sodio", "measureUnit": "mmol/L" }] </pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> Se non ci sono esami: “Attenzione! Nessun esame presente “ |

| | |
|---|---|
| Nome | getIncludedTypes |
| URL | GET - /atticus/api/instrumentalTest/getIncludedTypes/140 |
| Recupera la lista dei tipi degli esami strumentali associati all’assistito. | |
| Risposta | |
| STATUS 200 | <pre> application/json Restituisce i tipi degli esami strumentali: return List<InstrumentalTestTypeDTO> { "id": "1", </pre> |

| | |
|------------|--|
| | <pre>"name": "Ecg" }</pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> Se non ci sono esami: Attenzione! Nessun esame presente |

| | |
|---|---|
| Nome | addExam |
| URL | POST - /atticus/api/bloodTest/addExam |
| Inserisce un nuovo esame ematochimico di un assistito | |
| Input | <pre>{ "editDisabled":true, "inputError":false, "idType":1, "value":"20", "examType": {"id":1,"name":"Creatinina","measureUnit":"mg/dl"}, "date":"2020-04-23T03:00:00.000Z", "idPerson":"140" }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce il successo del salvataggio: return boolean</p> |

| | |
|------------|---|
| | <pre>{ True }</pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> • Se il salvataggio non dovesse andare a buon fine oppure se uno dei campi inseriti non è valido o non popolato: “Attenzione! Impossibile salvare gli esami ematochimici”. • Se non è stato possibile recuperare l’utente che ha inoltrato la richiesta: “Attenzione! Utente non trovato” |

| | |
|---|--|
| Nome | getInstrumentalTests |
| URL | POST - /atticus/api/instrumentalTest/getInstrumentalTests |
| Recupera la lista degli esami strumentali | |
| Input | <pre>{ "paginationAndSortingDetails":{ "pageNumber":0, "pageSize":5, "sortedBy":"id", "desc":false }, "filters":{ "personId":"140", </pre> |

| | |
|------------|--|
| | <pre> "idExamType":1 } } </pre> |
| Risposta | |
| STATUS 200 | <pre> application/json Restituisce l'esame strumentale salvato: return Page<InstrumentalTestDTO> { "content": {}, "0:{id: 45, date: \"2020-04-24T02:00:00.000+0000\",...}": {}, "1:{id: 46, date: \"2020-04-20T03:00:00.000+0000\",...}": {}, "2:{id: 71, date: \"2020-04-15T02:00:00.000+0000\",...}": {}, "pageable:{sort: {sorted: true, unsorted: false, empty: false}, offset: 0, pageNumber: 0, pageSize: 5,...}": {}, "sort:{sorted: true, unsorted: false, empty: false}": {}, "offset:0": {}, "pageNumber:0": {}, "pageSize:5": {}, "unpaged:false": {}, "paged:true": {}, "totalElements:3": {}, "last:true": {}, "totalPages:1": {}, "size:5": {}, </pre> |

| | |
|------------|--|
| | <pre>"number:0": {}, "numberOfElements:3": {}, "first:true": {}, "empty:false": {} }</pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> “Attenzione! Esame ematochimico non trovato”. “Attenzione! Tipo di esame ematochimico non trovato” |

| | |
|---|--|
| Nome | getInstrumentalTestsToComparing |
| URL | POST /atticus/api/instrumentalTest/getInstrumentalTestsToComparing |
| Recupera la lista degli esami ematochimici possibili da confrontare | |
| Input | <pre>{ "idType":1, "personId":"140" }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce gli esami strumentali: return List<InstrumentalTestDTO></p> |

```
{[
  {
    "id": "45",
    "date": "2020-04-24T02=00=00.000+0000",
    "fileUrl": "/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 24-04-2020 04-00-00.jpg",
    "idType": "1",
    "report": "null",
    "idUser": "2",
    "idPerson": "140",
    "user": "null",
    "examType": "null"}
  {
    "id": "71",
    "date": "2020-04-15T02=00=00.000+0000",
    "fileUrl": "/ESAMI_STRUMENTALI/DLEFNC96H11B519F/Ecg 15-04-2020 04-00-00.jpg",
    "idType": "1",
    "report": "null",
    "idUser": "2",
    "idPerson": "140",
    "user": "null",
    "examType": "null"
  }
}]
```

| | |
|------------|---|
| | } |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> Se l'esame strumentale non è presente: "Attenzione! Tipo di esame strumentale non trovato" |

| | |
|---|---|
| Nome | getAppointment |
| URL | GET - /atticus/api/appointment/getAppointment/{appointmentId} |
| Recupera le informazioni di un determinato appuntamento in base ad uno specifico identificativo idAppuntamento. | |
| Input | <pre>{ "idAppuntamento":1, "personId":"140" }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce l'Appuntamento: return AppointmentDTO</p> <p>Esempio: /atticus/api/appointment/getAppointment/15</p> <pre>{ "id": 15, "note": "Nuovo", "date": "2020-05-15T07:20:00.000+0000",</pre> |

| | |
|------------|--|
| | <pre>"idPerson": 1, "idUser": 2, "takePlace": 0 }</pre> |
| STATUS 500 | <p>AtticusException:</p> <ul style="list-style-type: none"> Se l'appuntamento non è presente all'interno de sistema: "Attenzione! Appuntamento non trovato" |

| | |
|---|--|
| Nome | addAppointment |
| URL | POST - /atticus/api/appointment/addAppointment |
| Aggiunge un nuovo appuntamento, dopo una serie di controlli di veridicità, obbligatorietà e unicità | |
| Input | <pre>{ "note": "Portare le ultime analisi", "date": "2020-05-15T07:20:00.000Z", "idPerson": 1 }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce il successo del salvataggio nel database: return ResponseOperation</p> <pre>{</pre> |

| | |
|------------|--|
| | <pre>"success": true }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e riporta la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e riporta la lista dei campi non validi con messaggio “Campo non valido” • Se almeno uno dei campi obbligatori non è popolato e almeno uno dei campi non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio”/“Campo non valido” • Se non si riesce a risalire all’utente loggato che sta tentando di aggiungere il nuovo appuntamento: “Attenzione! Utente non trovato” |

| | |
|---|--|
| Nome | findAppointmentsWithPagination |
| URL | POST - /atticus/api/appointment/findAppointmentsWithPagination/{lastTenAppointment} |
| <p>Recupera la lista paginata degli ultimi 10 appuntamenti dell’assistito presenti nel database o degli appuntamenti dell’assistito richiesti in base ai filtri inseriti.</p> | |

| | |
|------------|--|
| Input | <pre>{ "paginationAndSortingDetails": { "pageNumber": 0, "pageSize": 5, "sortedBy": "date" }, "filters": { "dateFrom": "2020-05-10", "dateTo": "2020-05-20", "personId": 1 } }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json</p> <p>Restituisce la lista paginata degli Utenti: return Page<AppointmentDTO></p> <pre>{ "content": [{ "id": 15, "note": "Nuovo", "date": "2020-05-15T07:20:00.000+0000", "idPerson": 1, }] }</pre> |

| | |
|--|--|
| | <pre> "idUser": 2, "takePlace": 0 }, { "id": 14, "note": "Nuovo", "date": "2020-05-18T20:15:00.000+0000", "idPerson": 1, "idUser": 2, "takePlace": 0 }, { "id": 18, "note": "Nuovo", "date": "2020-05-19T07:20:00.000+0000", "idPerson": 1, "idUser": 2, "takePlace": 0 }], "pageable": { "sort": { "sorted": true, "unsorted": false, </pre> |
|--|--|

| | |
|------------|---|
| | <pre> "empty": false }, "offset": 0, "pageNumber": 0, "pageSize": 5, "paged": true, "unpaged": false }, "totalPages": 1, "last": true, "totalElements": 3, "size": 5, "number": 0, "sort": { "sorted": true, "unsorted": false, "empty": false }, "numberOfElements": 3, "first": true, "empty": false } </pre> |
| STATUS 500 | AtticusException |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Se non è stato inserito alcun filtro: “Attenzione! Selezionare almeno un campo per effettuare la ricerca” e la lista dei campi vuoti con messaggio “Campo obbligatorio” • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore |
|--|--|

| | |
|---|--|
| Nome | getCheckUpAppointment |
| URL | GET - /atticus/api/checkUp/getCheckUpAppointment/{appointmentId} |
| Recupera le informazioni del Check-Up relativo all’Appuntamento, sulla base dell’id Appuntamento. | |
| Input | <pre>{ "id": 15, "note": "Nuovo", "date": "2020-05-15T07:20:00.000+0000", "idPerson": 1, "idUser": 2, "takePlace": 0 }</pre> |
| Risposta | |
| STATUS 200 | application/json |

| | |
|------------|---|
| | <p>Restituisce il Check-Up: return CheckUpDTO</p> <p>Esempio: /atticus/api/checkUp/getCheckUpAppointment /13</p> <pre>{ "id": 4, "idPerson": 1, "date": "2020-05-12T00:00:00.000+0000", "weight": 75, "height": 180, "sbpSupine": 30, "dbpSupine": 30, "sbpOrtostatic": 30, "dbpOrtostatic": 30, "spO2": 30, "glycemia": 30, "dyuresis": 30, "idUser": 2, "idAppointment": 13 }</pre> |
| STATUS 500 | AtticusException: |

| | |
|--|--|
| | <p>Se l'appuntamento non è stato ancora effettuato: "Attenzione! Appuntamento non ancora effettuato, non è stato trovato nessun Follow-Up relativo all'appuntamento"</p> <p>Esempio di eccezione:</p> <pre>{ "errorCode": " CHECK_UP_APPOINTMENT_NOT_EXIST ", "message": "Attenzione! Appuntamento non ancora effettuato, non è stato trovato nessun Follow-Up relativo all'appuntamento", "fieldsError": null }</pre> |
|--|--|

| | |
|-------|---|
| Nome | updateAppointment |
| URL | PUT /atticus/api/appointment/updateAppointment |
| | Aggiorna nel database l'appuntamento modificato, dopo una serie di controlli di veridicità, obbligatorietà e unicità |
| Input | <pre>{ "id": 15, "note": "Nuovo", "date": "2020-05-15T07:20:00.000+0000", "idPerson": 1, "idUser": 2,</pre> |

| | |
|------------|--|
| | <pre>"takePlace": 0 }</pre> |
| Risposta | |
| STATUS 200 | <pre>application/json Ritorna il successo del salvataggio nel database: return ResponseOperation { "success": true }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e la lista dei campi vuoti con messaggio “Campo obbligatorio” e campo che ha generato l’errore • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore • Se uno dei campi obbligatori non è popolato e uno non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio” / “Campo non valido” e campo che ha generato l’errore • Se l’appuntamento è già stato effettuato “Non è possibile modificare l'appuntamento” |

3.5 Servizi gestione monitoraggio on demand

| | |
|---|--|
| Nome | requestMonitoringOnDemand |
| URL | POST /atticus/api/monitoringOnDemand/requestMonitoringOnDemand |
| Richiede il monitoraggio on demand di un particolare assistito. | |
| Input | { "startDate": "2020-04-20T07:54:27.000+0000" } |
| Risposta | |
| STATUS 200 | application/json Ritorna il successo del salvataggio nel database: return ResponseOperation { "success": true } application/json – Restituisce la il log di monitoraggio della richiesta return LogMonitoringDTO { "id": 109, "idUser": 3, "idPerson": 159, "startDate": 1603458002562, |

| | |
|-------------------|--|
| | <pre> "endDate": null, "state": 0, "person": { "id": 159, "ssn": "DLEFNC96H11B519F", "name": "Francesco", "surname": "D'Elia", "birthDate": 1603097524000, "healthCode": "1", "idMonitoringKit": 60, "idMonitoringManager": 3, "registrationDate": 1603103506000, "removalDate": null, "monitoringState": "1", "idCaregiver": 4, "accessToData": 0, "idCaregiverRelationship": 1, "allowReporting": "0", "idDevice": "406d9bfe-8e92-4858-93b3-c1e6c65bf436", "externalPerson": null } } </pre> |
| <p>STATUS 500</p> | <p>AtticusException</p> <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato: “Attenzione! Campi non popolati” e la lista dei campi vuoti con messaggio “Campo obbligatorio” e campo che ha generato l’errore • Se uno dei campi non è popolato correttamente: “Attenzione! Campi non popolati correttamente” e la lista dei campi non validi con messaggio “Campo non valido” e campo che ha generato l’errore |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Se uno dei campi obbligatori non è popolato e uno non è popolato correttamente: “Attenzione! Campi obbligatori non popolati o non popolati correttamente” e la lista dei campi non popolati e non validi con messaggio “Campo obbligatorio” / “Campo non valido” e campo che ha generato l’errore • Se l’appuntamento è già stato effettuato “Non è possibile modificare l'appuntamento” |
|--|---|

| | |
|--|--|
| Nome | allowMonitoring |
| URL | POST /atticus/api/monitoringOnDemand/allowMonitoring |
| Gestione richiesta di Monitoraggio On-Demand: accetta richiesta, rifiuta richiesta o richiesta in timeout. | |
| Input | <pre>{ "idRequest": 51, "allow": "1" → [0 = rifiutata, 1 = accettata, 2 = timeout] "startingTime": null → popolato solo se allow = 1 }</pre> |
| Risposta | |
| STATUS 200 | application/json – return ResultDTO |

| | |
|------------|--|
| | <pre>{ "result": true, "resultCode": "ok", "resultMessage": "Richiesta gestita con successo" }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> Se il salvataggio della richiesta fallisce: "Attenzione! Si è verificato un problema con il salvataggio del log di monitoraggio" |

| | |
|---|---|
| Nome | allowMonitoring |
| URL | <p>GET</p> <p>/atticus/api/monitoringOnDemand/recordOnlineDataOnDemand/{idRequest}</p> |
| Registra Dati Monitoraggio su richiesta (json Online) | |
| Input | <pre>{ "_id" : ObjectId("5f92a2e40e80b90d3bce6569"), "idAlarm" : null, "kitId" : 60, "time" : NumberLong(1603445474495), "indoorArea" : "0", "breathRate" : 16, "heartRate" : 73, }</pre> |

| | |
|--|--|
| | <pre>"temperature" : 36.24, "gsr" : 9.52, "ecg" : [{ "lead" : "V1", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... 0.0554519420307769, 0.0509268023301723, 0.0467390743614896,], }, { "lead" : "V3", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... -0.0515732853256376,</pre> |
|--|--|

| | |
|--|---|
| | <pre> -0.06947869261007, -0.0911244539637506,] }, { "lead" : "V5", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, 0.208840355492592, 0.196020640691023] }, { "lead" : "RA", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, </pre> |
|--|---|

| | |
|--|---|
| | <pre> 0.208840355492592, 0.196020640691023] }, { "lead" : "LA", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... 0.456341373021469, 0.450111637084022, 0.443109318052442,] }], "breathingWave" : [0.584957429206583, 0.526950386996702, 0.466375649890296, ... 0.345634048669811, 0.410254968115528, </pre> |
|--|---|

| | |
|--|--|
| | <pre> 0.472876819690914,], "inertialData" : { "acc" : { "x" : [1.01527782720901, 1.02638896227504, 1.02638896227504, 1.01944450285877, 1.02222228662528, 1.01944450285877], "y" : [-0.12777778582816, -0.12777778582816, -0.12777778582816, ... -0.134722233088466, -0.130555557438704, -0.12777778582816], "z" : [0.0986111170596459, </pre> |
|--|--|

| | |
|--|---|
| | <pre>0.0916666697993398, 0.0916666697993398, ... 0.084722225390337, 0.0819444448505077, 0.080555590452358] , "gyro" : { "x" : [1.01527782720901, 1.02638896227504, 1.02638896227504, 1.01944450285877, 1.0222228662528, 1.01944450285877], "y" : [-0.12777778582816, -0.12777778582816, -0.12777778582816, ... -0.134722233088466,</pre> |
|--|---|

| | |
|--|--|
| | <pre>-0.130555557438704, -0.127777778582816], "z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ... 0.0847222225390337, 0.0819444448505077, 0.0805555590452358] }, "magn" : { "x" : [30.615233, 30.615233, 30.615233, ... 37.353514, 37.353514, 37.353514], "y" : [30.615233, 30.615233, 30.615233, ... 37.353514, 37.353514, 37.353514], "z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ... 0.0847222225390337, 0.0819444448505077, 0.0805555590452358] }</pre> |
|--|--|

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|-------------------|--|
| | <pre> 27.978514, 27.978514, 27.978514, ... 21.826171, 21.826171, 21.826171], "z" : [30.615233, 30.615233, 30.615233, ... 27.392577, 27.392577, 27.392577] } }, } </pre> |
| Risposta | |
| STATUS 200 | <pre> application/json – return ResultDTO { </pre> |

| | |
|------------|--|
| | <pre>"result": true, "resultCode": "ok", "resultMessage": "Richiesta accettata" }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> Se il salvataggio dei dati fallisce: "Attenzione! Si è verificato un problema con il salvataggio dei dati di monitoraggio online (realtime)" |

| | |
|--|--|
| Nome | requestMonitoringOnDemand |
| URL | GET atticus/api/online/get |
| Recupera le informazioni del monitoraggio. | |
| Input | <pre>{ "idKit": 60, "since": 65467571 }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json – Restituisce la lista dei dati del monitoraggio (Json Online)</p> <p>return List<Online></p> <pre>[</pre> |

```
{
  "_id" : ObjectId("5f92a2e40e80b90d3bce6569"),
  "kitId" : 60,
  "time" : NumberLong(1603445474495),
  "breathRate" : 16,
  "heartRate" : 73,
  "temperature" : 36.24,
  "gsr" : 9.52,
  "ecg" : [
    {
      "lead" : "V1",
      "data" : [
        0.0497083856780602,
        0.0422363069106421,
        0.0347467695542567,
        ...,
        0.0554519420307769,
        0.0509268023301723,
        0.0467390743614896,
      ],
    },
    {
      "lead" : "V3",
```

| | |
|--|---|
| | <pre>"data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... -0.0515732853256376, -0.06947869261007, -0.0911244539637506,] }, { "lead" : "V5", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, 0.208840355492592, 0.196020640691023] }, {"lead" : "RA", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, 0.208840355492592, 0.196020640691023] }</pre> |
|--|---|

```
0.0497083856780602,  
0.0422363069106421,  
0.0347467695542567,  
....  
0.221414332777715,  
0.208840355492592,  
0.196020640691023  
]  
,  
{  
  "lead" : "LA",  
  "data" : [  
    0.0497083856780602,  
    0.0422363069106421,  
    0.0347467695542567,  
    ...  
    0.456341373021469,  
    0.450111637084022,  
    0.443109318052442,  
  ]  
}  
],  
"breathingWave" : [  
  0.584957429206583,
```

```

0.526950386996702,
0.466375649890296,
...
0.345634048669811,
0.410254968115528,
0.472876819690914,
],
"inertialData" : {
  "acc" : {
    "x" : [
      1.01527782720901,
      1.02638896227504,
      1.02638896227504,
      ....
      1.01944450285877,
      1.02222228662528,
      1.01944450285877
    ],
    "y" : [
      -0.12777778582816,
      -0.12777778582816,
      -0.12777778582816,
      ...
      -0.134722233088466,

```

| | |
|--|--|
| | <pre> -0.130555557438704, -0.127777778582816], "z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ... 0.0847222225390337, 0.0819444448505077, 0.0805555590452358] }, "gyro" : { "x" : [1.01527782720901, 1.02638896227504, 1.02638896227504, 1.01944450285877, 1.02222228662528, 1.01944450285877], "y" : [</pre> |
|--|--|

| | |
|--|---|
| | <pre>-0.12777778582816, -0.12777778582816, -0.12777778582816, ... -0.134722233088466, -0.130555557438704, -0.12777778582816], "z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ... 0.084722225390337, 0.0819444448505077, 0.0805555590452358] }, "magn" : { "x" : [30.615233, 30.615233, 30.615233, ...</pre> |
|--|---|

| | |
|--|---|
| | <pre>37.353514, 37.353514, 37.353514 , "y" : [27.978514, 27.978514, 27.978514, ... 21.826171, 21.826171, 21.826171 , "z" : [30.615233, 30.615233, 30.615233, ... 27.392577, 27.392577, 27.392577] } },</pre> |
|--|---|

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|----|
| | }} |
|--|----|

3.6 Servizi gestione allarmi

| | |
|--|---|
| Nome | takeAlarms |
| URL | GET /atticus/api/alarm/notifications/takeAlarms |
| <p>Recupera le informazioni sugli allarmi in corso sulla base dell'utente loggato (il medico specialista o responsabile del monitoraggio otterranno solo le notifiche di allarme relative ai proprio assistiti). Questo servizio non restituisce gli allarmi bianchi, poiché il rischio che questi siano falsi positivi è troppo alto.</p> | |
| Risposta | |
| STATUS 200 | <p>application/json – Restituisce la lista delle notifiche di allarme</p> <p style="text-align: center;">return List<AlarmNotificationsDTO></p> <pre>[{ "id": 100, "idAlarm": 50, "idUser": 3, "idPerson": 159, "notified": 1,</pre> |

175

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| | |
|--|--|
| | <pre>"person": { "id": 159, "ssn": "DLEFNC96H11B519F", "name": "Francesco", "surname": "D'Elia", "birthDate": 1603097524000, "healthCode": "1", "idMonitoringKit": 60, "idMonitoringManager": 3, "registrationDate": 1603103506000, "removalDate": null, "monitoringState": "1", "idCaregiver": 4, "accessToData": 0, "idCaregiverRelationship": 1, "allowReporting": "0", "idDevice": "406d9bfe-8e92-4858-93b3-c1e6c65bf436", "externalPerson": null }, "alarm": { "id": 50, "idMonitoringKit": 60, "idUser": null, "alarmDate": 1603712347000,</pre> |
|--|--|

| | |
|--|---|
| | <pre> "latitude": 42.2328, "longitude": 12.3424, "type": 2, "state": 1, "diagnosis": null, "anomalySource": "Cuore", "anomalyDetected": null, "inDoorArea": "1", "person": null } }]</pre> |
|--|---|

| | |
|--|---|
| Nome | <i>deleteAlarmNotifications</i> |
| URL | PUT /alarm/notifications/delete/{idAlarm} |
| Elimina le notifiche di allarme sulla base dell'id allarme, dopo che uno degli utenti ha preso in carica l'allarme e salva l'id dell'utente che lo sta gestendo. | |
| Input | <pre> { "id": 15, "note": "Nuovo", "date": "2020-05-15T07:20:00.000+0000", </pre> |

| | |
|------------|---|
| | <pre>"idPerson": 1, "idUser": 2, "takePlace": 0 }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json – return ResultDTO</p> <p>Esempio: http://localhost:9090/atticus/api/alarm/notifications/delete/5</p> <pre>{ "result": true, "resultCode": "ok", "resultMessage": "" }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se l'allarme è già stato preso in carico: "Attenzione! Allarme già preso in carico da un altro medico" • Se il salvataggio dell'id utente è fallito: "Attenzione! Si è verificato un problema con il salvataggio dei dati dell'utente che ha preso in carica l'allarme" • Se l'eliminazione delle notifiche di allarme è fallita: "Attenzione! Si è verificato un problema con l'eliminazione delle notifiche di allarme" |

| | |
|------|--------------------|
| Nome | <i>recordAlarm</i> |
| URL | PUT |

| | |
|---|--|
| | atticus/api/alarm/recordAlarm |
| Registra un allarme e crea le notifiche di allarme nel caso di allarme rosso, giallo o grigio | |
| Input | <pre>{ "idKit": 60, "type": 1, "lat": 42.2328354, "lng": 12.3424324, "anomalySource": "Cuore", "inDoorArea": "1" }</pre> |
| Risposta | |
| STATUS 200 | <p>application/json – return ResultDTO</p> <p>Esempio: http://localhost:9090/atticus/api/alarm/recordAlarm</p> <pre>{ "result": true, "resultCode": "ok", "resultMessage": "Allarme registrato" }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se il salvataggio dell'allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio dell'allarme" • Se il salvataggio delle notifiche di allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio delle notifiche di allarme" |

| | |
|--------------------------------------|--|
| Nome | <i>recordWarning</i> |
| URL | PUT atticus/api/alarm/recordWarning |
| Registra un warning ed avvisa il DSS | |
| Input | { "iDKit": 60, "type": 2, "lat": 42.2328354, "lng": 12.3424324, "anomalySource": "Cuore", "inDoorArea": "1" } |
| Risposta | |
| STATUS 200 | application/json – return ResultDTO Esempio: http://localhost:9090/atticus/api/alarm/recordWarning { "result": true, "resultCode": "ok", "resultMessage": "Warning registrato" } |

| | |
|--|---|
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> Se non vengono recuperate le soglie di allarme: “Attenzione! Si è verificato un problema con il recupero delle soglie di allarme” |
| Nome | <i>confirmAlarm</i> |
| URL | <p>PUT</p> <p>atticus/api/alarm/confirmAlarm</p> |
| <p>Conferma un allarme e crea le notifiche di allarme nel caso di allarme rosso, giallo o grigio</p> | |
| Input | <pre>{ "idAlarm": 60, "anomaly": "Cuore", "type": 1 }</pre> |
| <p>Risposta</p> | |
| STATUS 200 | <p>application/json – return ResultDTO</p> <p>Esempio: http://localhost:9090/atticus/api/alarm/confirmAlarm</p> <pre>{ "result": true, "resultCode": "ok", "resultMessage": "Allarme confermato" }</pre> |

| | |
|------------|--|
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se non viene recuperato l'allarme: "Attenzione! Allarme non trovato" • Se il salvataggio dell'allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio dell'allarme" • Se il salvataggio delle notifiche di allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio delle notifiche di allarme" – |
|------------|--|

| | |
|--|--|
| Nome | <i>rejectAlarm</i> |
| URL | PUT atticus/api/alarm/rejectAlarm/{idAlarm} |
| Rigetta un allarme sulla base dell'id dell'allarme | |
| Risposta | |
| STATUS 200 | <p>application/json – return ResultDTO</p> <p>Esempio: http://localhost:9090/atticus/api/alarm/rejectAlarm/7</p> <pre>{ "result": true, "resultCode": "ok", "resultMessage": "Allarme rigettato" }</pre> |
| STATUS 500 | <p>AtticusException</p> <ul style="list-style-type: none"> • Se non viene recuperato l'allarme: "Attenzione! Allarme non trovato" • Se il salvataggio dell'allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio dell'allarme" |

| | |
|--|---|
| Nome | <i>manageAlarm</i> |
| URL | PUT atticus/api/alarm/manageAlarm |
| Gestisci Allarme: Conferma Allarme o Rifiuta Allarme al seconda del valore del campo “action” passato in input | |
| Input | { "alarmId": 60, "monitoringKitId": 3, "diagnosis": "Aritmia cardiaca", "action": true (true = conferma; false = rifiuta) } |
| Risposta | |
| STATUS 200 | application/json – return ResultDTO Esempio: http://localhost:9090/atticus/api/alarm/confirmAlarm { "result": true, "resultCode": “ok”, "resultMessage": “Allarme confermato” } |
| STATUS 500 | AtticusException <ul style="list-style-type: none"> Se non viene recuperato l’allarme: “Attenzione! Allarme non trovato” |

| | |
|--|--|
| | <ul style="list-style-type: none"> Se il salvataggio dell'allarme fallisce: "Attenzione! Si è verificato un problema con il salvataggio dell'allarme" |
|--|--|

| | |
|---|--|
| Nome | <i>recordOnLineData</i> |
| URL | PUT atticus/api/online/record |
| Registra Dati Monitoraggio su richiesta (json Online) | |
| Input | <pre>{ "_id" : ObjectId("5f92a2e40e80b90d3bce6569"), "idAlarm" : 5, "kitId" : 60, "time" : NumberLong(1603445474495), "indoorArea" : "0", "location" : { "lat" : 42.2328354, "lng" : 12.3424324 }, "breathRate" : 16, "heartRate" : 73, "temperature" : 36.24, "gsr" : 9.52, "ecg" : [</pre> |

| | |
|--|---|
| | <pre>{ "lead" : "V1", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... 0.0554519420307769, 0.0509268023301723, 0.0467390743614896,], }, { "lead" : "V3", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, ... -0.0515732853256376, -0.06947869261007, -0.0911244539637506,] }</pre> |
|--|---|

| | |
|--|--|
| | <pre>}, { "lead" : "V5", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, 0.208840355492592, 0.196020640691023] }, { "lead" : "RA", "data" : [0.0497083856780602, 0.0422363069106421, 0.0347467695542567, 0.221414332777715, 0.208840355492592, 0.196020640691023] }</pre> |
|--|--|

```
    },  
    {  
      "lead" : "LA",  
      "data" : [  
        0.0497083856780602,  
        0.0422363069106421,  
        0.0347467695542567,  
        ...  
        0.456341373021469,  
        0.450111637084022,  
        0.443109318052442,  
      ]  
    }  
  ],  
  "breathingWave" : [  
    0.584957429206583,  
    0.526950386996702,  
    0.466375649890296,  
    ...  
    0.345634048669811,  
    0.410254968115528,  
    0.472876819690914,  
  ],  
  "inertialData" : {
```

| | |
|--|---|
| | <pre>"acc" : { "x" : [1.01527782720901, 1.02638896227504, 1.02638896227504, 1.01944450285877, 1.02222228662528, 1.01944450285877], "y" : [-0.12777778582816, -0.12777778582816, -0.12777778582816, ... -0.134722233088466, -0.130555557438704, -0.12777778582816], "z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ...] }</pre> |
|--|---|

| | |
|--|---|
| | <pre> 0.0847222225390337, 0.08194444448505077, 0.0805555590452358] }, "gyro" : { "x" : [1.01527782720901, 1.02638896227504, 1.02638896227504, 1.01944450285877, 1.02222228662528, 1.01944450285877], "y" : [-0.12777778582816, -0.12777778582816, -0.12777778582816, ... -0.134722233088466, -0.130555557438704, -0.12777778582816], </pre> |
|--|---|

| | |
|--|---|
| | <pre>"z" : [0.0986111170596459, 0.0916666697993398, 0.0916666697993398, ... 0.0847222225390337, 0.0819444448505077, 0.0805555590452358] , "magn" : { "x" : [30.615233, 30.615233, 30.615233, ... 37.353514, 37.353514, 37.353514], "y" : [27.978514, 27.978514, 27.978514, ... 34.545454, 34.545454, 34.545454] }</pre> |
|--|---|

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|-------------------|---|
| | <pre> ... 21.826171, 21.826171, 21.826171], "z" : [30.615233, 30.615233, 30.615233, ... 27.392577, 27.392577, 27.392577] } }, } </pre> |
| Risposta | |
| STATUS 200 | <pre> application/json – return List<AnomalyMapDTO> { "alarmId": 5, "type": 1, "source": "" } </pre> |

191

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|--|---|
| | } |
|--|---|

4 Servizi del DSS di livello 3

In questa sezione si descrivono i servizi esposti dal DSS di livello 3 al software di monitoraggio e i servizi esposti dai *detector* implementati come moduli esterni e richiamati dal DSS di livello 3. Il DSS di livello 3, inoltre, esporrà un'interfaccia utilizzabile dagli altri DSS (DSS di livello 2 o 1) per segnalare avvisi di anomalia e iniziare un monitoraggio in tempo reale.

| | |
|---|---|
| Nome | healthcheck |
| URL | GET - /atticus/dss/healthcheck |
| Definisce lo stato di salute del DSS, verrà chiamato periodicamente per determinare se il DSS è online oppure offline | |
| Input | Vuoto |
| Risposta | |
| STATUS 200 | <i>text/html</i> Restituisce un messaggio di controllo "I'm alive" |
| STATUS 500 | <i>text/html</i> Restituisce un messaggio di errore "Internal Server Error" |

| | |
|---|----------------------------|
| Nome | detect |
| URL | POST - /atticus/dss/detect |
| Richiama la detection degli algoritmi predittivi sulla base di un avviso di anomalia dato in input, che riguarda uno specifico assistito. Gli algoritmi predittivi che verranno richiamati dipenderanno dal tipo di anomalia indicata (es. cuore o respiro) | |

| | |
|------------|---|
| Input | <p><i>application/json</i></p> <p><i>JSON che descrive, per uno specifico paziente, l'avviso di anomalia su cui lanciare la detection</i></p> <pre>{ "kitId": 1, "alarmId": 2, "alarmTimestamp": 1602689472, "anomalySource": "Heart" }</pre> |
| Risposta | |
| STATUS 200 | <p><i>application/json</i></p> <p><i>Restituisce un json che descrive l'esito della richiesta</i></p> <pre>{ "result": true, "resultCode": "200", "resultMessage": "OK" }</pre> |
| STATUS 400 | <p><i>text/html</i></p> <p><i>Restituisce un messaggio di errore dato da un JSON di input non valido</i></p> <p>"One or more fields are not valid or missing"</p> |
| STATUS 500 | <p><i>text/html</i></p> <p><i>Restituisce un messaggio di errore</i></p> <p>"Internal Server Error"</p> |

4.1 Servizi dei moduli esterni al DSS

I moduli esterni al DSS comunicano con una con un'interfaccia comune tramite REST API. L'interfaccia prevede l'implementazione dei due servizi specificato di seguito.

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|---|---|
| Nome | healthcheck |
| URL | GET - /atticus/detector/<NOME_DETECTOR>/healthcheck |
| Definisce lo stato di salute del Detector, verrà chiamato periodicamente per determinare se il Detector è online oppure offline | |
| Input | Vuoto |
| Risposta | |
| STATUS 200 | <p><i>application/json</i></p> <p><i>Restituisce un json che descrive l'esito della richiesta</i></p> <pre>{ "result": true, "resultCode": "200", "resultMessage": "OK" }</pre> |
| STATUS 500 | <p><i>text/html</i></p> <p><i>Restituisce un messaggio di errore</i></p> <p>"Internal Server Error"</p> |

| | |
|--|--|
| Nome | predict |
| URL | POST - /atticus/detector/<NOME_DETECTOR>/predict |
| Richiama la predizione del modello predittivo sulla base di dati inviati in input. | |
| Input | <p><i>application/json</i></p> <p><i>JSON che contiene i dati da dare in input al modello predittivo.</i></p> <pre>{</pre> |

195

Progetto finanziato nell'ambito del programma PON "Ricerca e Innovazione 2014 e 2020" D.D. del 02.08.2018 Prot. n. 2056

| | |
|------------|--|
| | <pre>"kitId": "1", "alarmId": "1", "fs": 128, "data": [0.26813, 0.38479, 0.32811, 0.21005, -0.016679, -0.1768, ..., -0.89894] }</pre> |
| Risposta | |
| STATUS 200 | <p><i>application/json</i></p> <p><i>Restituisce un json che descrive l'esito della richiesta</i></p> <pre>{ "kitId": string, "alarmId": string, "prediction": List[int] }</pre> |
| STATUS 422 | <p><i>application/json</i></p> <p><i>Restituisce un messaggio di errore dato da un JSON di input non valido. Nel campo "detail" sarà contenuta la descrizione dell'errore, inclusi i campi ritenuti errati.</i></p> <pre>{ "detail": [{ "loc": ["body", "request", "idAlarm"], "msg": "field required", "type": "value_error.missing" }] }</pre> |

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

| | |
|------------|---|
| STATUS 500 | <p><i>application/json</i></p> <p><i>Restituisce un messaggio di errore</i></p> <pre>{ "detail": "prediction error" }</pre> |
|------------|---|

5 Struttura del DSS di livello 2

Il DSS di livello 2, similmente a quello di livello 3, ha una struttura modulare. In Figura 13 è rappresentato il diagramma dei componenti del DSS di livello 2. Il modulo Main si occupa della gestione del processo di detection da parte degli algoritmi predittivi, mentre il modulo Base contiene tutti gli elementi condivisi. Per ogni algoritmo predittivo, ci sarà un apposito modulo Detector specializzato nell'analisi di un'anomalia specifica. Per il DSS di livello 2 è previsto il rilascio sotto forma di libreria Java, da integrare sia nella S-BOX che nell'App MyATTICUS. Per alcuni detector si è fatta eccezione, per esempio il detector di tracking fuori casa, che verrà sviluppato specificatamente a seconda della piattaforma di destinazione. Questo è dovuto al fatto che sono richieste librerie specifiche che dipendono dalla piattaforma (es. Tensorflow, librerie per GPS).

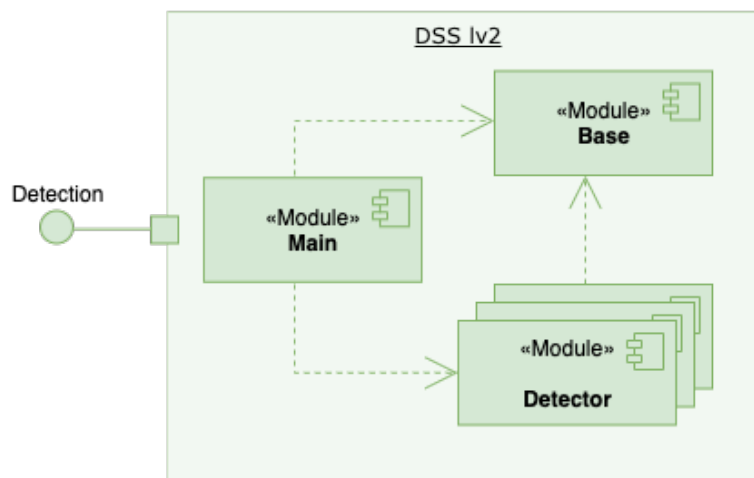


Figura 13. Struttura del DSS di livello 2

6 Struttura dell'App MyATTICUS

In questa sezione verrà descritto il funzionamento e la struttura dell'App MyATTICUS, che verrà eseguita su smartphone ed integrerà la ricezione dei dati dallo smart wearable, l'elaborazione degli stessi e il DSS di livello 2.

6.1 Architettura dell'App MyATTICUS

L'App MyATTICUS sarà presente solo sullo smartphone e avrà principalmente la funzione di lettura e processamento dei dati. Nel momento in cui il dispositivo smart wearable non sarà più in comunicazione con la S-BOX, l'App MyATTICUS avrà il compito di processare i dati di monitoraggio ed effettuare le rilevazioni delle anomalie tramite il DSS di livello 2. Il primo prototipo di MyATTICUS verrà rilasciato per la piattaforma Android. In Figura 14 viene rappresentato il component diagram di MyATTICUS. Ricevuti in input i dati di monitoraggio, il servizio di lettura da Bluetooth Low Energy (*BLE Service*) si occuperà di leggere i dati raw ricevuti e convertirli per il processamento. Ci sarà il modulo relativo al DSS di livello 2 (*DSS lv2*) e uno o più detector esterni (*ExternalDetector*) che eseguiranno le analisi sui dati ricevuti. Nel caso specifico di Android, i detector esterni potranno utilizzare le librerie relative a *Google Maps* per la geolocalizzazione e *Tensorflow Lite* per gli algoritmi predittivi. Il tutto è coordinato dal modulo principale (*MainActivity*) che si occuperà di restituire sull'interfaccia grafica informazioni sullo stato di funzionamento del monitoraggio (es. attivo/non attivo) e in seguito l'invio dei dati di monitoraggio e di eventuali allarmi e warning al software di monitoraggio tramite il modulo REST Controller.

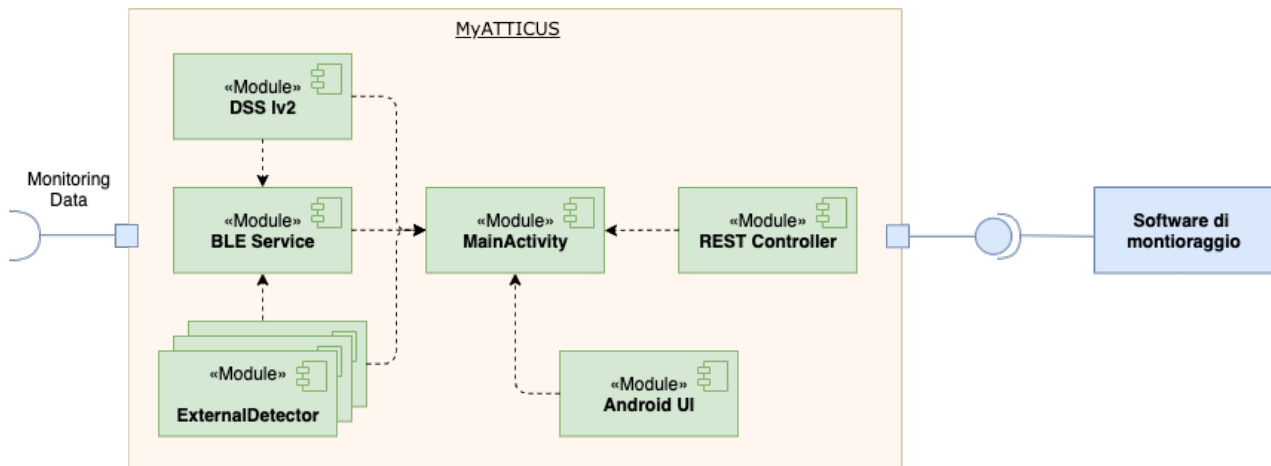


Figura 14. Struttura dell'App MyATTICUS

6.2 Rilevazione delle anomalie nell'App MyATTICUS

La rilevazione delle anomalie nell'App MyATTICUS viene effettuata, usando gli algoritmi contenuti nel DSS di livello 2 e i detector esterni, sui dati ricevuti in input dallo smart wearable. La rilevazione è continua, e inizia non appena viene effettuato l'accesso a MyATTICUS e viene configurato il kit di monitoraggio. In Figura 15 sono mostrate le schermate di login e la homepage dell'App. Alla prima installazione, è necessario effettuare l'accesso utilizzando il codice fornito con il kit di monitoraggio. Successivamente, per permettere la rilevazione di posizione *in-door* o *out-door*, è necessario indicare la posizione della propria abitazione/domicilio (Figura 16). A questo punto MyATTICUS sarà configurata per l'utilizzo, dove sarà presente una schermata di riepilogo (Figura 15) che mostrerà informazioni sullo stato attuale del monitoraggio e altre informazioni relative al kit. Un esempio è la barra di stato che indica se il dispositivo wearable è attualmente connesso o disconnesso dallo smartphone.



Figura 15. Schermata di Login (sinistra) e schermata home (destra) dell'App MyATTICUS

Deliverable D6.3

Ambient-intelligent Tele-monitoring and Telemetry for Incepting & Catering over Human Sustainability

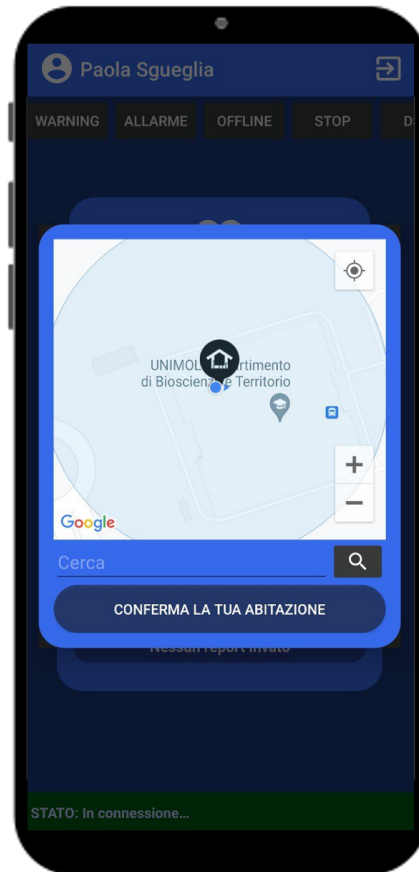


Figura 16. Impostazione della posizione dell'abitazione su MyATTICUS

7 Sicurezza del sistema

In questa sezione sono presentati gli aspetti tecnologici e le strategie adottate per garantire sia la sicurezza della comunicazione sia la sicurezza dei dati.

7.1 Architettura secure-by-design e by-default

I concetti di *security by design* e *security by-default* sono stati introdotti con la nuova normativa europea relativa ai dati, ovvero il GDPR, e riguardano, rispettivamente, i principi di protezione dei dati sin dal momento della progettazione e l'impostazione predefinita dei dati relativi ai mezzi utilizzati per il trattamento dei dati personali, con lo scopo di tutelare i principi fondamentali in materia di privacy introdotti dal GDPR.

Con *data protection by design* bisogna prevedere, già in fase di progettazione, meccanismi di protezione degli stessi, mentre, nel caso di *data protection by default*, bisogna limitare il trattamento dei dati solo a specifiche finalità e per un tempo strettamente necessario. Sia per il *security by design* che per il *security by-default* bisogna garantire l'adozione delle misure di tecniche idonee a proteggere i dati personali degli interessati nei sistemi informatici e nelle infrastrutture di rete utilizzate per il trattamento.

7.1.1 Conformità al GDPR (UE 2016/679)

I principi sopra citati richiedono procedure pratiche che impattano direttamente sia sul processo di sviluppo che sulle decisioni prese in fase di progettazione. Entrambi fanno parte dell'Art. 25 [1] del GDPR. Verranno riportate, per ognuno, le parti di interesse.

Nel caso di *security by design*, l'articolo prescrive:

<< Tenendo conto dello stato dell'arte e dei costi di attuazione, nonché della natura, dell'ambito di applicazione, del contesto e delle finalità del trattamento, come anche dei rischi aventi probabilità e gravità diverse per i diritti e le libertà delle persone fisiche costituiti dal trattamento, sia al momento di determinare i mezzi del trattamento sia all'atto del trattamento stesso il titolare del trattamento mette in atto misure tecniche e organizzative adeguate, quali la pseudonimizzazione, volte ad attuare in modo efficace i principi di protezione dei dati, quali la minimizzazione, e a integrare nel trattamento le necessarie garanzie al fine di soddisfare i requisiti del presente regolamento e tutelare i diritti degli interessati >>

Dall'estratto dall'articolo, come anticipato, si evince che, nel momento in cui bisognerà determinare le metodologie del trattamento dei dati, quindi dalla fase iniziale alla successiva concretizzazione, bisogna prevedere misure adeguate atte a rispettare i principi di protezione dei dati. Un'applicazione concreta, nel caso di ATTICUS, è l'utilizzo di tecniche di *pseudonimizzazione* descritte nella Sezione 5.4. Inoltre, sono previste ulteriori misure, quali la presenza di un sistema RBAC per la gestione degli accessi (Sezione 5.2) e l'utilizzo di protocolli di comunicazione sicuri (Sezione 5.3).

Per quanto riguarda il principio di *security by default*, l'articolo prescrive:

<<Il titolare del trattamento mette in atto misure tecniche e organizzative adeguate per garantire che siano trattati, per impostazione predefinita, solo i dati personali necessari per ogni specifica finalità del trattamento. Tale obbligo vale per la quantità dei dati personali raccolti, la portata del trattamento, il periodo di conservazione e l'accessibilità. In particolare, dette misure garantiscono che, per impostazione predefinita, non siano resi accessibili dati personali a un numero indefinito di persone fisiche senza l'intervento della persona fisica. >>

In questo caso, in ATTICUS verranno trattati solo i dati strettamente legati al monitoraggio dell'assistito e i dati necessari alla figura del medico specialista o del responsabile di monitoraggio. Un esempio è la rilevazione della posizione dell'assistito che verrà mostrata solo in caso di monitoraggio in tempo reale dovuto a una situazione di allarme. Inoltre, l'utilizzo del sistema RBAC (Sezione 5.2) garantisce l'accesso ai dati dell'assistito alle sole figure autorizzate.

7.1.2 Standard di riferimento (ISO 27001 e 27034)

La serie ISO/IEC 27000 [2] è una raccolta di linee guida (standard) introdotte da un comitato di esperti con lo scopo di definire le norme riguardanti i Sistemi di Gestione per la Sicurezza delle Informazioni (SGSI). Nello specifico, le norme della serie definiscono i requisiti per creare un SGSI, le linee guida di progettazione ed evoluzione e, inoltre, ne valutano la conformità. Le norme inoltre sono interdipendenti tra loro e possono essere suddivise in diverse macroaree, ovvero:

- *norme che descrivono una panoramica e la tecnologia;*
- *norme di specifica dei requisiti;*
- *norme che indicano le linee guida generali;*
- *norme che descrivono linee guida per settori o ambiti specifici di applicazione.*

Le norme in questione, invece, riguardano i requisiti per impostare e gestire un sistema SGSI inclusi aspetti relativi alla sicurezza logica, fisica ed organizzativa (ISO 27001) e le linee guida per la sicurezza informatica (ISO 27034).

In particolare, lo standard ISO 27001 riguarda non solo la tutela della privacy, come definito dal GDPR, ma anche la tutela dei dati di business che riguardano l'organizzazione. Infatti, le informazioni sono considerate come un asset aziendale e, quindi, devono essere protette da eventuali minacce per garantire la continuità del business con un'impostazione di prevenzione dei rischi ad esse collegati. Lo standard ISO 27001 può essere considerato come una esemplificazione del GDPR,

poiché fornisce linee guida alla sicurezza delle informazioni che vanno dalla protezione degli stessi fino alla riduzione di minacce informatiche, mentre non vengono coperti tutti gli aspetti relativi alla privacy dei dati. I requisiti per la certificazione ISO 27001 [3] prevedono la documentazione e quindi la definizione di differenti aspetti del SGSI, come ad esempio scope del sistema (deliverable D1.1, D1.2, D1.3, D1.4, D1.5, D1.6) le politiche di sicurezza (D2.5) e risk assessment (D2.1). Lo standard ISO 27034 [4], invece, è suddiviso in più parti e fornisce le linee guida da adottare durante il processo di progettazione e sviluppo in termini di sicurezza delle informazioni, che vanno dalla protezione delle informazioni che vengono utilizzate da un'applicazione fino alla protezione delle stesse da accessi non autorizzati. Questi aspetti, nel caso di ATTICUS, sono correttamente trattati come descritto dalle sezioni che seguono (Sezioni. 7.3 e 7.4).

7.2 Autenticazione dei servizi

I servizi offerti dal back-end vengono utilizzati sia dal front-end che dal DSS. Dato che sono necessariamente pubblici, verrà implementato un meccanismo di autenticazione basato su Role Based Access Control (RBAC) e JSON Web Token (JWT).

7.2.1 Role Based Access Control (RBAC)

Il meccanismo di RBAC prevede la definizione di *ruoli* che permettono di limitare l'accesso alle informazioni ad utenti non autorizzati. I ruoli sono collegati a specifici utenti, che hanno l'autorizzazione ad accedere a un insieme specifico di transazioni.

Principalmente, il meccanismo di RBAC si basa su tre regole fondamentali:

- *Assegnazione dei ruoli*: ad ogni transazione devono essere assegnati uno o più ruoli che possono eseguire tale transazione;

- *Autorizzazione dei ruoli:* un ruolo attivo per un determinato utente deve essere stato autorizzato per quell'utente;
- *Autorizzazione alla transazione:* un utente può eseguire una transazione solo se la transazione è autorizzata per quel ruolo attivato per quell'utente.

Nel caso dei servizi, verranno definiti dei ruoli per ogni tipologia di attore che utilizzerà il front-end (es. *caregiver* o medico specialista). Inoltre, verranno creati dei ruoli per il DSS di livello 3 e per MyATTICUS. Questo farà sì che, ad esempio, il DSS potrà accedere solo ai servizi di *conferma* e *rigetto* di un warning, mentre si impedirà l'accesso a tali funzionalità all'applicazione MyATTICUS, al fine di garantire l'integrità del sistema.

7.2.2 JSON Web Token (JWT)

L'autenticazione, dal punto di vista pratico, sarà gestita mediante token JWT. Si tratta di uno standard utilizzato per cifrare, autenticare un utente ed eventualmente garantire, tramite firma, l'autenticità dello stesso. Un JWT è un JSON composto principalmente da tre parti:

- *Header:* indica la tipologia di algoritmi usati per la firma e la cifratura;
- *Payload:* contiene una serie di "attestazioni" e dati necessari all'autenticazione. Con attestazione si intende, ad esempio, che il token identifica un utente autenticato come "admin" ad una determinata data (*timestamp*);
- *Signature:* la firma che viene utilizzata per validare il token.

Tutti gli attori del software di monitoraggio, inclusi i sistemi quali DSS di livello 3 e MyATTICUS, dovranno richiamare il servizio di autenticazione fornito dal back-end per effettuare una richiesta ad un servizio protetto. Il servizio, date le credenziali di accesso associate allo specifico utente, verificherà la correttezza delle stesse e, se l'esito del controllo è positivo, fornirà in output il token JWT che dovrà essere usato per l'autenticazione nelle successive richieste ai servizi. Ad esempio, un

dato client richiede un token con un ruolo di “*caregiver*”. Il server genera il token richiesto da un client se questo fornisce le credenziali corrette. Da quel momento in poi, il client userà il token per autenticarsi come “*caregiver*” e, quindi, accedere alle risorse richieste.

7.3 Sicurezza della comunicazione

Il sistema ATTICUS può contare su una comunicazione sicura tra il front-end e il back-end grazie all’utilizzo del protocollo HTTPS (*Hypertext Transfer Protocol Secure*). Il protocollo HTTPS, a differenza del protocollo HTTP, crea un canale di comunicazione sicuro tra chi effettua la chiamata e chi la riceve utilizzando come primitiva crittografica principale la cifratura. Grazie a tale meccanismo, un potenziale attaccante che riuscisse ad intromettersi nella comunicazione tra client e server intercettandone i “messaggi”, non potrebbe in alcun modo comprenderne il contenuto, poiché potrà leggere solo la versione cifrata della comunicazione, decifrabile solo se si è a conoscenza della chiave. In questo modo, ogni richiesta ed ogni risposta della comunicazione è perfettamente confidenziale.

Inoltre, come ulteriore misura di sicurezza, è stato predisposto un isolamento della rete dei container Docker. Le uniche componenti esposte a chiamate esterne alla suddetta rete sono quelle che devono essere pubbliche per permettere il funzionamento del sistema, ovvero il container del front-end, che deve essere richiamabile per raggiungere il livello di presentazione del sistema, e quello del back-end, che deve offrire i suoi servizi al front-end. Il container che ospita il database, invece, può essere raggiungibile solo dai container interni alla rete e, quindi, non sarà visibile all’esterno.

7.4 Sicurezza dei dati

Come visto in precedenza, il sistema tratta, oltre ai dati anagrafici degli assistiti, anche i loro dati sanitari. Per tale motivo, sono state valutate tutte le precauzioni necessarie a garantire la riservatezza di tali informazioni così da soddisfare la GDPR compliance. In primo luogo, come previsto dal GDPR, è stata adottata una misura di pseudonimizzazione di tutti i dati personali e sensibili. Per “pseudonimizzazione” si intende un processo che va a mascherare le informazioni rendendole apparentemente non identificative. A differenza dell’anonimizzazione, che rende questo mascheramento irreversibile, la pseudonimizzazione permette di riottenere le informazioni originali che possono essere ancora utilizzate dal sistema. La tecnica impiegata, nel dettaglio, è quella della cifratura mediante algoritmo AES (Advanced Encryption Standard) che rappresenta attualmente, come indica il nome stesso, lo standard in tale contesto. Ciò significa che allo stato attuale non esiste un attacco pratico allo schema se non un attacco di forza bruta (*brute force attack*) atto a trovare la chiave di decifratura provando tutte le combinazioni possibili di caratteri, operazione che potrebbe impiegare per una chiave a 128 bit (16 caratteri) più tempo della presunta età dell’universo.

Sebbene il livello di sicurezza sarebbe stato potenzialmente già alto limitandosi alla cifratura dei dati, come visto in precedenza si è provveduto ad implementare un’ulteriore strategia che riguarda la separazione del database contenente i dati anagrafici dell’assistito dal database che contiene i dati sanitari dello stesso. I dati nei due database sono legati attraverso una chiave che identifica univocamente un assistito all’interno del sistema ATTICUS.

Nello specifico, nel database dei dati anagrafici l’assistito è identificato da un codice criptato con chiave C1, mentre nel database dei dati sanitari lo stesso utente è identificato sempre dallo stesso codice, ma, questa volta, criptato con chiave C2. Ciò significa che le due stringhe risultanti sono

diverse. Quindi, qualora un utente malizioso riuscisse a effettuare un *data breach*, rubando entrambi i database, per riuscire a legare i dati anagrafici ai dati sanitari dovrebbe decifrare due codici, criptati con due chiavi distinte. Pertanto, se anche si riuscisse ad ottenere una chiave di cifratura, senza l'altra non sarebbe comunque possibile effettuare l'incrocio delle informazioni. Considerando già la difficoltà di fondo di acquisire le informazioni contenute in un database, tenendo conto della cifratura e considerando il tempo necessario a derivare una chiave per AES, questa ulteriore misura garantisce un elevatissimo livello di sicurezza.

Va, inoltre, osservato che i dati sanitari del paziente, all'interno del database dei dati sanitari, sono memorizzati in maniera anonima. Nessun dato anagrafico, infatti, è contenuto in tale database e il codice univoco che identifica l'assistito non contiene al suo interno alcun riferimento a dati anagrafici del paziente (quali, ad esempio, il codice fiscale).

Ovviamente, il back-end del software di monitoraggio, in possesso di entrambe le chiavi, può effettuare l'incrocio dei dati anagrafici con i dati sanitari, restituendo tali informazioni (decifrate) a chi richiama i suoi servizi ed è autorizzato a farlo.

Come specificato in D2.5, i dati di monitoraggio conservati in MongoDB saranno cifrati a loro volta attraverso un'ulteriore chiave di cifratura, utilizzando sempre l'algoritmo AES.

Riferimenti bibliografici

- [1] PrivazyPlan (2020), *EU regolamento generale sulla protezione dei dati (EU-RGPD)*. Da <https://www.privacy-regulation.eu/it/index.htm>.
- [2] Wikipedia (2020). Da https://it.wikipedia.org/wiki/ISO/IEC_27000-series
- [3] ISECT, *ISO/IEC 27001 (2020)*. Da <https://www.iso27001security.com/html/27001.html>.
- [4] ISECT, *ISO/IEC 27034 (2020)*. Da <https://www.iso27001security.com/html/27034.html>.