

**ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA**

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

**INTERNET OF MEDICAL THINGS E SVILUPPO DI SISTEMI
INTEROPERABILI BASATI SU STANDARD FHIR: UN CASO
DI STUDIO BASATO SULL'INTEGRAZIONE DI UN
DISPOSITIVO EGA**

Tesi di laurea in
SISTEMI EMBEDDED E INTERNET OF THINGS

Relatore

Prof. ALESSANDRO RICCI

Candidato

EDOARDO MAIOLI

Correlatore

Dott. Ing. ANGELO CROATTI

Anno Accademico 2020-2021

The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.
(Kevin Ashton)

Indice

Introduzione	xiii
1 Internet of Medical Things	1
1.1 Internet of Things: overview e applicazioni	2
1.2 IoT and IoMT system Architecture	4
1.2.1 Perception Layer o Sensor Layer	5
1.2.2 Network Layer	7
1.2.3 Service Layer	9
1.2.4 Application Layer	11
1.3 IoMT Main Applications	12
1.3.1 Monitoraggio della frequenza cardiaca	12
1.3.2 Rilevamento del livello di glucosio nel sangue	13
1.3.3 Monitoraggio della Temperatura corporea	13
1.3.4 Monitoraggio della saturazione dell'ossigeno nel sangue	14
1.3.5 Monitoraggio della pressione sanguigna	14
1.3.6 Sistemi di riabilitazione	14
1.3.7 Ambulanza intelligente	15
1.3.8 Altre applicazioni	15
1.3.9 Ospedale intelligente	16
1.4 Sicurezza IoMT	17
1.4.1 Simmetric-Key Algorithms	18
1.4.2 Asymmetric-Key Algorithms	18
1.4.3 Keyless Algorithms	19

2	Attuali standard di interoperabilità e FHIR	21
2.1	Integrating the Healthcare Enterprise (IHE)	22
2.2	ISO/IEEE 11073	22
2.2.1	ISO/IEEE 11073 10201	23
2.3	HL7	24
2.3.1	HL7 2.x	24
2.3.2	HL7 3.x	25
2.4	HL7 Fast Healthcare Interoperability Resources (FHIR)	26
2.4.1	Struttura e risorse	27
2.4.2	Profili ed estensioni	28
2.4.3	API RESTful	29
2.4.4	Ricerca	30
2.4.5	Point-of-Care Device Implementation Guide	30
2.5	openEHR	32
2.5.1	Obiettivi openEHR	32
2.5.2	Applicazione openEHR COVID-19	32
3	Emogasanalisi e caso di studio	35
3.1	Cos'è	35
3.2	Dispositivi EGA	35
3.2.1	RAPIDPoint 500 Siemens	36
3.3	Analisi dominio e risorse	38
3.3.1	Device	39
3.3.2	DeviceMetric	39
3.3.3	Observation e DiagnosticReport	39
3.3.4	Diagramma risorse FHIR del dominio	40
3.4	Modellazione Risorse PoCD Implementation Guide	40
3.4.1	MDS e VMD, top level Device	41
3.4.2	Channel	42
3.4.3	DeviceMetric e Observation	42
3.4.4	Applicazione del modello	43

4	Progettazione Componente FHIR per il progetto T4C	47
4.1	Tracking for Care	47
4.2	Integrazione con il componente proposto	48
4.2.1	FHIR Server	50
4.2.2	Requisiti del componente	50
4.2.3	T4C e Componente	50
4.3	Progettazione	52
4.4	Sviluppo e Testing	53
4.4.1	Tecnologie utilizzate	55
4.4.2	Testing	56
4.4.3	Approcci differenti	59

Elenco delle figure

1.1	Architettura a 3 layers[31]	5
1.2	Ambulanza intelligente[11]	15
1.3	Ospedale Intelligente[24]	17
2.1	Standard core[20]	23
2.2	Struttura di una risorsa	27
2.3	A sinistra i livelli dei profili dal core a quelli organizzativi, a destra il restringimento delle risorse conformi.	28
3.1	Dispositivo EGA RAPIDPoint 500	36
3.2	Sensori con tecniche di misurazione	37
3.3	Diagramma generico del dominio	38
3.4	Diagramma FHIR del dominio	40
3.5	MDS e VMD	41
3.6	Channels	42
3.7	Acido-base	43
3.8	Co-ossimetria	44
3.9	Elettroliti	44
3.10	Metaboliti	45
3.11	Ossigenazione	45
3.12	Esempio Observation	45
4.1	Diagramma di sequenza stato attuale	49
4.2	Diagramma di sequenza con EgaComponent	51
4.3	Progettazione ideale	52

4.4	Risorse FHIR	53
4.5	Sistema complessivo	54

Elenco dei listati

2.1	Esempio di messaggio semplificato di HL7 v3	26
4.1	Esempio di DiagnosticReport aggiunto al FHIR Server	56
4.2	Esempio di risposta alla chiamata API relativa a tutti Device	58

Introduzione

Ci troviamo in un mondo in costante cambiamento ed evoluzione, in particolar modo nel settore informatico. È proprio in questo continuo processo di evoluzione che spicca l'IoT ovvero Internet Of Things (Internet delle cose). Oggetti, anche di tutti i giorni, collegati ad Internet capaci di scambiarsi messaggi e di comunicare tra loro. Kevin Ashton, pioniere dell'IoT, dice che questa sarà la prossima rivoluzione tecnologica come lo è stato a suo tempo Internet. Tuttavia, come sarà discusso in questa tesi, alcuni ambiti di applicazione dell'IoT come il sistema medico ed ospedaliero non sono ancora del tutto uniformati ma anzi, presentano vari ostacoli nell'ottica di un sistema interoperabile ed intelligente.

Per quanto riguarda quest'ultimo caso si parla di Internet of Medical Things (IoMT), Internet of Healthcare Things (IoHT) o ancora Medicina 4.0 (Quarta Rivoluzione Industriale in ambito medico). Possiamo pensare ad un ospedale con vari dispositivi capaci di condividere dati ed informazioni tra loro e di mettere a disposizione questi dati in tempo reale al personale sanitario e al paziente anche a distanza. In tal senso è opportuno definire due termini importanti; telemedicina e telemonitoraggio. L'enciclopedia Treccani [34] definisce la telemedicina come:

”l'insieme delle tecniche e degli strumenti di monitoraggio e di assistenza sanitaria, realizzato mediante sistemi atti a fornire un rapido accesso sia ai medici specialisti che ai pazienti, prescindendo dal luogo ove essi sono rispettivamente situati. In altri termini, la t. consente di fornire un'assistenza medica adeguata a pazienti fisicamente distanti, mediante l'adozione congiunta delle tecnologie dell'informatica e delle telecomunicazioni, le quali costituiscono, assieme, la cosiddetta 'telematica' ”

Invece con telemonitoraggio si intende, in telemedicina, il monitoraggio a di-

stanza da parte di una apparecchiatura che lo consente.

L'introduzione di questi device "intelligenti" punta a migliorare in termini di velocità e di efficacia i servizi sanitari, diminuire la mole di lavoro del personale e a tagliare i costi delle spese mediche. Si tratta quindi di un processo che inizia nel rilevamento dei dati relativi ai pazienti sia remoti che fisici da parte dei sensori di questi device intelligenti, i dati possono poi passare attraverso un Gateway che permette la connessione ad un server o ad un cloud service per la processazione dei dati. Infine i dati vengono resi disponibili a tutti i dispositivi che ne hanno bisogno tramite le opportune interfacce. Un esempio possono essere i cosiddetti EHR (ELECTRONIC HEALTH RECORDS) come il Fascicolo Sanitario Elettronico, che permettono al paziente di accedere ai propri dati sanitari in remoto nel momento in cui questi vengono resi accessibili. La destinazione dei dati non è detto che sia sempre il paziente, basti infatti pensare, sempre in ambito della telemedicina, il caso in cui il paziente sia direttamente in cura da remoto con dispositivi intelligenti. In questo caso i medici e il personale sanitario possono gestire e monitorare i dati in real time, ovvero nel momento in cui vengono raccolti e in base a questi reagire di conseguenza. Ogni giorno si presentano nuove sfide nell'ambito dell'IoMT e nuovi progetti estremamente interessanti.

Nella primo capitolo di questa tesi mi occuperò di effettuare, partendo dall'IoT e dalla sua architettura, uno studio approfondito sullo stato dell'arte dell'Internet of Medical Things , sulle sue ultime innovazioni e su quelle future, sugli standard utilizzati e sui dispositivi che attualmente ne fanno uso. Nel secondo capitolo si entrerà più nello specifico su quali sono gli attuali standard di interoperabilità, con particolare attenzione per HL7 FHIR. Nel terzo capitolo si parlerà dell'emogasanalisi, del dispositivo RAPIDPoint 500 situato all'ospedale Bufalini di Cesena e di come modellare tale dispositivo tramite risorse FHIR. Nel terzo capitolo verrà analizzato il caso di studio Tracking for Care per il dispositivo in questione e verrà proposta una soluzione che si integri al progetto in un ottica interoperabile.

Capitolo 1

Internet of Medical Things

Quando parliamo di IoT parliamo di intelligenza ed evoluzione. Si tratta di un mondo di oggetti interconnessi tramite Internet capaci di elaborare e scambiare dati. È in questo contesto che si inserisce Internet of Medical Things (IoMT), ovvero l'applicazione dell'IoT al sistema medico ed ospedaliero. Si aprono quindi infinite prospettive sul potenziale che può avere l'IoT in un campo così vasto. Partiamo con un po' di dati. La multinazionale Gartner che si occupa di consulenza strategica e analisi nel campo della tecnologia dell'informazione nel 2019, aveva previsto che nel 2020 ci sarebbero stati 5.8 miliardi di dispositivi IoT connessi, ora sappiamo che ce ne sono stati 11.7 (secondo il sito statista.com). Secondo l'articolo "Recent Advances in the Internet-of-Medical-Things"[14] l'industria del settore sanitario potrebbe risparmiare fino a 300 miliardi di dollari facendo affidamento sui device IoMT, specialmente per le malattie croniche e per la telemedicina. E ancora, si stima che il mercato dell'IoMT avrà un valore di 135 miliardi di dollari entro il 2025.

È importante, prima di addentrarci nelle sue applicazioni, fornire un contesto e una breve descrizione della situazione odierna dell'IoMT.

Siamo tutt'ora molto lontani da un sistema ideale dove ogni apparecchio medico è capace di collegarsi ad Internet e di inviare i propri dati in un formato standard verso un web server o un cloud, di modo da renderli accessibili al personale sanitario e ai pazienti in qualsiasi momento. Abbiamo invece una varietà di apparecchi medici che non possono collegarsi alla rete e quelli che lo fanno, non

sono interoperabili con apparecchi di altri vendors. Quello che succede il più delle volte è che il personale sanitario stampa direttamente i dati raccolti e li mantiene in formato cartaceo.

In questo capitolo si proseguirà dapprima facendo un quadro generale dell'IoT poi si proseguirà descrivendo le principali applicazioni dell'IoT nel settore sanitario per poi passare all'architettura di un IoMT system e finire con un rapido excursus sul lato security.

1.1 Internet of Things: overview e applicazioni

L'Internet delle cose (IoT) è un argomento importante nell'industria tecnologica ed informatica ed è diventato un titolo di cronaca sia sulla stampa specializzata che sui media popolari. Questa tecnologia si concretizza in un ampio spettro di prodotti, sistemi e sensori collegati in rete, che sfruttano i progressi in potenza di calcolo, miniaturizzazione dell'elettronica e interconnessioni di rete per offrire nuove capacità non precedentemente possibili. Un'abbondanza di conferenze, rapporti e articoli di giornale discutono e dibattono sul potenziale impatto della "rivoluzione IoT": da nuove opportunità di mercato e modelli di business alle preoccupazioni su sicurezza, privacy e interoperabilità tecnica. L'implementazione su larga scala dei dispositivi IoT promette di trasformare molti aspetti del nostro modo di vivere. Per i consumatori, nuovi prodotti IoT come elettrodomestici abilitati a Internet, componenti di automazione domestica e dispositivi di gestione energetica ci stanno spostando verso una visione della "casa intelligente", che offre maggiore sicurezza ed efficienza energetica. Altri dispositivi IoT personali come i dispositivi indossabili per il monitoraggio della salute e del fitness e i dispositivi medici abilitati alla rete stanno trasformando il modo in cui vengono forniti i servizi sanitari. Questa tecnologia promette di essere vantaggiosa per le persone con disabilità e gli anziani, consentendo maggiori livelli di indipendenza e qualità della vita a un costo ragionevole. Sistemi IoT come veicoli capaci di collegarsi alla rete, traffico intelligente, sistemi e sensori incorporati in strade e ponti ci avvicinano all'idea di "città intelligenti", che aiutano a ridurre al minimo la congestione e il consumo di energia. La tecnologia IoT offre la possibilità di trasformare agricoltu-

ra, industria e produzione e distribuzione di energia aumentando la disponibilità di informazioni lungo la catena di produzione utilizzando sensori in rete. Tuttavia, l'IoT solleva molti problemi e sfide che devono essere considerati e affrontati al fine di realizzare potenziali benefici. Se anche le più modeste previsioni sono corrette, un'esplosione delle applicazioni IoT potrebbe portare ad un cambiamento fondamentale nel modo in cui gli utenti interagiscono con Internet e da come ne sono influenzati, sollevando nuovi problemi e diverse sfide non solo nel campo tecnologico. Nonostante il concetto di combinare i computer e le reti per monitorare o controllare dei dispositivi non sia nuovo, il termine "Internet of Things" (IoT) è relativamente recente. Esso è stato utilizzato per la prima volta nel 1999 dal pioniere della tecnologia britannica Kevin Ashton per descrivere un sistema in cui gli oggetti nel mondo fisico potrebbero essere collegati a Internet tramite sensori. Ashton ha coniato il termine per illustrare il potere di collegare i tag di identificazione a radiofrequenza (RFID), utilizzati nelle catene di approvvigionamento aziendali, a Internet per contare e tenere traccia delle merci senza la necessità di un intervento umano. Oggi, Internet of Things è diventato un termine popolare per descrivere scenari in cui la connettività ad Internet e la capacità di elaborazione si estendono a una varietà di oggetti, dispositivi, sensori ed elementi di tutti i giorni[29]. Le potenzialità offerte dall'IoT rendono possibile la sua applicazione in vari ambiti, di cui solo una piccolissima parte è attualmente a disposizione della nostra società. Molti sono i domini e gli ambienti in cui queste nuove applicazioni migliorerebbero la qualità della vita. L'obiettivo è dare a questi oggetti la possibilità di comunicare tra loro ed elaborare le informazioni percepite dall'ambiente circostante. Si possono identificare 4 domini applicativi dell'IoT[7]:

- Dominio dei trasporti e logistica
- Ambiente intelligente (casa, ufficio, città, ecc.)
- Dominio personale e sociale
- Dominio Healthcare

Per quanto riguarda il dominio dei trasporti si parla di auto tecnologicamente avanzate (guida assistita e autonoma), treni, autobus e biciclette che stanno diventando

sempre più strumentate con sensori e attuatori. Le strade stesse e le merci trasportate sono inoltre dotate di tag e sensori che inviano informazioni importanti ai siti di controllo del traffico e ai veicoli di trasporto per instradare meglio il traffico, aiutare nella gestione dei depositi, fornire al turista adeguate informazioni sui trasporti e monitorare lo stato delle merci in movimento. Un ambiente intelligente permette alle persone che lo occupano di sfruttarne tutte peculiarità facilmente e con comodità. Questi ambienti possono essere case ed uffici con ambienti riscaldati in base al tempo o alle preferenze individuali, stanze con illuminazione automatica in base all'orario, o ancora, sistemi di allarme per prevenire incidenti domestici. Le applicazioni del dominio sociale sono quelle che consentono all'utente di interagire con altre persone, costruire e mantenere relazioni sociali. In ottica IoT le "cose" intelligenti possono inviare automaticamente dei messaggi agli amici per consentire loro di sapere cosa stiamo facendo o dove stiamo andando, oppure per suggerire un certo percorso o un viaggio. In ultimo troviamo una delle applicazioni potenzialmente più importanti e di impatto dell'IoT nella società odierna, ovvero il dominio dell'Healthcare. I suoi ambiti applicativi verranno analizzati in dettaglio nella sezione 1.3.

1.2 IoT and IoMT system Architecture

Dato che l'IoMT segue gli stessi paradigmi architetturali dell'IoT, in questa sezione si presenteranno i tipi di architetture, i layer che le compongono e per ognuno di essi le tecnologie e i protocollo principali. Nell'IoT vi sono tre principali architetture: a 3, 4 e 5 layer o livelli.

L'architettura più semplice in assoluto è l'architettura a 3 livelli come mostrata in figura 1.2. È stata introdotta all'inizio delle ricerche in quest'ambito e prevede il livello di percezione (perception layer o sensor layer), di rete (network layer), e di applicazione (application layer) [6].

- Il livello di percezione è il livello fisico ed è caratterizzato dai sensori e dalla raccolta di dati e informazioni.

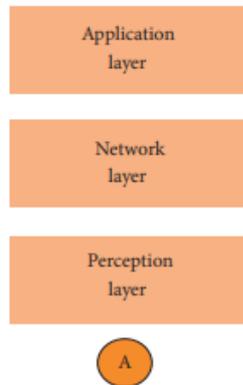


Figura 1.1: Architettura a 3 layers[31]

- Il livello di rete è responsabile per la connessione con altri dispositivi intelligenti, dispositivi di rete e server. Si occupa anche della trasmissione e dell'elaborazione dei dati.
- Il livello di applicazione si occupa di distribuire all'utente finale i servizi e le applicazioni necessarie per accedere ai dati in modo semplice ed efficace.

La semplicità di questa architettura nasconde funzioni ed operazioni diverse e complesse nei livelli di rete e applicazione. Per esempio, il livello di rete non solo ha bisogno di determinare il percorso e di trasmettere i dati ma fornisce anche dei servizi per questi come il data aggregation o computing. Il livello di applicazione, a sua volta, non deve solo fornire servizi ai clienti ed altri dispositivi, ma anche sui dati stessi, come il data mining o il data analytics. Per stabilire un'architettura generica e flessibile, è necessario un layer di mezzo o service layer, tra il network layer e l'application layer [23]. Questo livello si occuperà di fornire tutti i servizi sui dati.

Basandosi sull'architettura descritta si procede nel presentare le principali tecnologie e protocolli di ciascuno dei 4 livelli.

1.2.1 Perception Layer o Sensor Layer

Nel livello di percezione la funzione principale è identificare e tracciare gli oggetti. Per il completamento di questa funzione possono essere implementate le seguenti

tecnologie.

- **RFID.** L'RFID (Radio Frequency Identification) è una tecnologia di comunicazione contact-less e permette di identificare e tracciare gli oggetti senza un contatto fisico. Essa supporta lo scambio dei dati attraverso segnali radio su piccole distanze. Prevede 3 parti principali: RFID reader, RFID tag e l'antenna. Il tag RFID può essere semplicemente un microchip attaccato ad una antenna. Ogni tag RFID è associato ad un oggetto e ha il numero identificativo univoco. Un lettore RFID può identificare un oggetto, e ottenere le rispettive informazioni, interrogando il tag RFID con i giusti segnali. L'antenna viene usata per trasmettere i segnali dal tag al reader. Un tag RFID può essere passivo, attivo o BAP (battery assisted passive). Un tag passivo presenta un microchip non alimentato direttamente dalla corrente elettrica ma che al passaggio di un lettore, che lo inonda con frequenze radio, viene alimentato con l'energia necessaria per rispondere (il processo funziona anche in scrittura). Un tag RFID attivo è invece munito di una o più antenne e di una batteria integrata per l'alimentazione e questo gli permette di inviare in continuazione i propri segnali su frequenze diverse. In ultimo il tag BAP è simile a quello attivo ma invia le informazioni solo se interrogato da un reader. A confronto con le altre tecnologie, l'RFID presenta numerosi vantaggi: scan veloce, durabilità, dimensioni ridotte, basso costo, lettura contact-less, sicurezza e riusabilità. Un esempio di applicazione dell'RFID al mondo IoMT è presentato e spiegato in [21]. Viene infatti sfruttata questa tecnologia low cost ed estremamente semplice per creare una rete di sensori e microcontrollori capaci di leggere ed elaborare dati e permettere poi di inviare questi ultimi allo smartphone o al computer. Un altro esempio può essere [33] dove viene proposto un sistema capace di fornire le informazioni sanitarie di un paziente con l'aiuto di Internet e dei tag RFID. In questo caso i tag RFID creano un canale di comunicazione per il sistema delle informazioni sanitarie permettendo una specie di automazione amministrativa per quanto riguarda alcune azioni giornaliere come permessi e rimozione o pubblicazione di dettagli informativi.
- **NFC** La tecnologia Near Field Communication (NFC) non è altro che una

delle ultime applicazioni della tecnologia RFID. Nata nel 2002 da Philips e Sony permette una comunicazione contact-less, a corto raggio e half duplex tra due dispositivi compatibili. La comunicazione, che avviene entro pochi centimetri, viene effettuata ad una frequenza di 13.65 MHz. Il dispositivo che inizia la comunicazione viene chiamato "initiator" (l'iniziatore) mentre quello che risponde alla richiesta dell'iniziatore è il target. Questa tipologia di comunicazione è molto simile all'architettura client-server dove chi comincia la comunicazione è il client e il server risponde. In ottica NFC alcuni dispositivi potrebbero avere o non avere una fonte di energia interna. Si parla, rispettivamente, di dispositivi attivi e passivi. Chi inizia la comunicazione (l'iniziatore) deve sempre essere un dispositivo attivo, mentre chi risponde può essere attivo o passivo e, nel secondo caso, usare il campo elettromagnetico generato dal dispositivo attivo per rispondere. Vi sono tre dispositivi NFC: NFC enabled mobile phone, NFC reader e NFC tag. Il primo si riferisce semplicemente a quei dispositivi mobile che hanno l'NFC integrato. L'NFC reader è un lettore con un componente NFC. L'esempio più comune è il POS che permette di effettuare pagamenti istantanei e contact-less. L'NFC tag è semplicemente un RFID tag che non ha una fonte di energia integrata. L'NFC rappresenta uno dei protocolli di comunicazione wireless principali nel mondo IoT e, di conseguenza, anche in quello IoMT.

1.2.2 Network Layer

Il livello di rete o network layer svolge un ruolo fondamentale in questa architettura, si occupa di determinare i percorsi di routing e di fornire supporto adeguato alla trasmissione dei dati.

- **6LoWPAN** è l'abbreviazione per "IPv6 over Low Power Wireless Personal Area Network" ovvero IPv6 su reti a basso consumo wireless. Si tratta infatti della combinazione dei protocolli IPv6 e LoWPAN. È una rete a basso consumo ed ampio raggio che, supportata dagli standard IP, permette il collegamento diretto con Internet. È caratterizzata da un grande numero di dispositivi low cost connessi tramite comunicazione wireless. Grazie ai vantaggi della rete LoWPAN, ovvero ridotta dimensione dei pacchetti, basso

consumo e ridotta larghezza di banda, 6LoWPAN è direttamente applicabile nel mondo IoT e IoMT come rete di comunicazione. Per esempio in [26] 6LoWPAN viene proposta come rete per il monitoraggio da remoto delle condizioni di salute dei pazienti.

- **ZigBee** è una tecnologia wireless ideata per comunicazioni a basso consumo energetico e di breve durata. ZigBee include numerosi vantaggi come il basso consumo energetico, basso costo, semplicità, affidabilità e sicurezza. Per esempio in [13] viene discusso come i dati raccolti dai dispositivi medici come il monitor a parametri vitali o le pompe per infusione possono essere integrati all'interno del sistema ospedaliero sfruttando la tecnologia wireless di ZigBee. Quest'ultima infatti rispecchia perfettamente i requisiti di un sistema di questo tipo, in termini di larghezza di banda, potenza, sicurezza e mobilità.
- **Z-Wave** è una tecnologia di comunicazione wireless a breve durata con i vantaggi di essere a basso costo, basso consumo energetico ed estremamente affidabile. Il principale obiettivo di Z-Wave è quello di fornire una trasmissione affidabile tra un'unità di controllo e uno o più dispositivi. Segue infatti un'architettura master-slave dove il master (Z-wave Controller) controlla tutti gli slave (dispositivi o nodi che fanno parte della rete). Sebbene sia Z-Wave che ZigBee supportino comunicazione a corto raggio con costi di implementazione e consumi energetici bassi, queste due tecnologie presentano sostanziali differenze. Quella principale consiste nella frequenza di banda a cui operano a livello fisico; ZigBee opera normalmente a 2.4 GHz, mentre Z-Wave a meno di 1GHz. O ancora, la rete ZigBee può supportare fino a 65000 dispositivi mentre Z-Wave solo fino a 232, tuttavia quest'ultima risulta di più facile implementazione.
- **LoRaWAN** (Long Range Wide Area Network). LoRa è un protocollo di comunicazione wireless ideato appositamente per supportare reti a basso consumo energetico a livello globale, nazionale e regionale. Il raggio di trasmissione di LoRa arriva fino a 15 km. Questa tecnologia è composta da due parti: la parte fisica cosiddetta "LoRa" e i livelli superiori "LoRaWAN". Un

esempio di applicazione di questa rete viene proposto in [27], dove il caso di studio era, sempre in ambito sanitario, capire se data una sequenza di frames essa contenesse quelli di una persona in caduta. Viene proposto un framework che oltre che far uso di una Rete Neurale Convoluzionale (CNN) sfrutta il monitor a parametri vitali e i messaggi di emergenza per il personale sanitario.

1.2.3 Service Layer

Come già menzionato il Service Layer è situato tra il Network Layer e l'Application Layer e fornisce servizi sicuri ed efficienti agli oggetti o applicazioni.

- **Interface.** L'interfaccia permette una comunicazione sicura tra i dispositivi e le applicazioni e una gestione efficiente dei dispositivi interconnessi, includendo la connessione e disconnessione degli stessi. Per il supporto delle applicazioni un interface profile può essere considerato un servizio standard per facilitare l'interazione tra servizi forniti da dispositivi e applicazioni differenti.
- **Service Managment.** Il Service Managment si occupa di rilevare dispositivi e applicazioni, e pianificare servizi efficienti e affidabili per soddisfarne le richieste. Un servizio può essere considerato come un comportamento, inclusi la raccolta, lo scambio e memorizzazione dei dati, o un'associazione di questi comportamenti per raggiungere un obiettivo specifico. Nell'IoT alcuni requisiti possono essere soddisfatti da un solo servizio, mentre altri hanno bisogno di una loro integrazione. Si può suddividere il servizio in due categorie: servizio primario e secondario. Il servizio primario o servizio di base, espone le funzionalità primarie a dispositivi e applicazioni. Il servizio secondario implementa le funzionalità ausiliare in base ad un servizio primario o ad un altro servizio secondario.
- **Middleware.** Il Middleware è un software in grado di fornire un'astrazione tra tecnologie e applicazioni IoT. Nel middleware i dettagli delle diverse tecnologie sono nascosti e sono fornite le interfacce standard per consentire

agli sviluppatori di concentrarsi sullo sviluppo di applicazioni. Utilizzando un middleware, dispositivi e applicazioni con differenti interfacce possono scambiarsi informazioni e condividere risorse l'uno con l'altro. Negli ultimi anni sono stati proposti vari tipi di middleware.

- **Fog-Based Middleware.** Il Fog Computing è considerato da molte ricerche come un middleware tra l'IoT e il cloud computing. Questo strato intermedio riduce drasticamente la latenza e fornisce al settore medico la potenza di elaborazione e archiviazione richiesti. In effetti, molte situazioni sanitarie richiedono una reazione tempestiva. In sostanza si tratta di un'architettura simile al cloud computing dove però l'elaborazione dei data non è centralizzata nei cloud computers ma ne avviene una parte nei sensori e nei gateway di rete. Gli autori in [12] sostengono che l'uso del Fog Computing nel settore sanitario possa garantire che l'aggregazione dei dati dei pazienti venga fatta in rispetto della loro privacy e riservatezza.
- **Publish/Subscribe-Based Middleware.** Nel dominio medico-sanitario è necessario reagire tempestivamente ad una molteplicità di eventi e situazioni diverse. In questo torna utile il middleware di tipo Publish/Subscribe che affronta questi problemi proponendo un'architettura dinamica ed asincrona, dove i publisher forniscono le informazioni necessarie ai subscribers.
- **Web of Things-Based Middleware.** Si tratta di un middleware basato sul Web of Things, dove i dispositivi fisici vengono digitalizzati per permettere l'uso dei loro dati in applicazioni differenti. Per esempio, sensori del corpo che inviano i loro dati attraverso un'interfaccia web per poi essere salvati in un database relazionale con il supporto del cloud computing. Basandosi sul web questo middleware sfrutta le classiche tecnologie in quest'ambito, come Html e servizi REST.
- **SoA Based Middleware** Service oriented Architecture (SoA) si basa principalmente su servizi che offrono funzionalità pubbliche attraverso un'interfaccia che nasconde i dettagli interni. Il vantaggio principale di

questi middleware è la loro capacità di utilizzare più dispositivi garantendo loro una comunicazione completamente indipendente da qualsiasi architettura sottostante.

- **Message Oriented Middleware (MoM)** permette di scambiare un elevato numero di messaggi tra le applicazioni distribuite. Un MoM è proposto in [9] per unificare l'accesso a tutti i dispositivi medici al fine di facilitare la raccolta dei dati dai pazienti ai medici. In questo caso, per aumentare la sicurezza del middleware, è stata aggiunta un'applicazione RESTful per verificare quale gestore è autorizzato ad accedere a quali dati.

1.2.4 Application Layer

Nel livello di applicazioni ci si limiterà a descrivere brevemente i principali protocolli applicativi.

- **Advanced Message Queing Protocolo (AMQP)** è un protocollo standard di accodamento di messaggi utilizzato per fornire un servizio di messaggistica (accodamento, instradamento, sicurezza, affidabilità, ecc.). AMQP fornisce ai client una comunicazione stabile con i middleware anche se scritti in linguaggi diversi. Inoltre, AMQP implementa vari tipi di architetture di scambio di messaggi come store and forward, publish and subscribe, message distribution, message queing, context-based routing e instradamento point-to-point.
- **Constrained Application Protocolo (CoAP)** è un protocollo di messaggistica basato su architettura REST. Dato che molti dispositivi IoT hanno risorse limitate (poco spazio di archiviazione, bassa capacità computazionale, ecc.) il protocollo HTTP non dovrebbe essere usato data la sua complessità. Per superare il problema, è stato proposto CoAP, ovvero il protocollo a livello applicazione dello stack 6LoWPAN. L'obiettivo di CoAP è quello di consentire a dispositivi con risorse limitate di interagire con servizi RESTful. CoAP permette alcune importanti funzionalità, come il trasporto di risorse

a blocchi, la discovery delle risorse o l'osservazione delle stesse, linterazione con HTTP e la sicurezza.

- **Message Queing Telemetry Protocol (MQTT)** è un protocollo leggero con un codice semplice, il che lo rende facile da installare in dispositivi piccoli e limitati. Si tratta di un protocollo di comunicazione basato su TCP/IP. È stato ideato per comunicazioni remote con restrizioni di velocità (in tempo reale) e di efficienza energetica.
- **Extensible Messaging and Presence Protocol (XMPP)**. In ultimo è importante citare XMPP in quanto si tratta di un protocollo open source basato su XML. Creato per essere sicuro e scalabile presenta un modo semplice per effettuare l'indirizzamento dei dispositivi. Viene principalmente usato per le applicazioni di chatting, chiamate e video streaming.

1.3 IoMT Main Applications

L'IoMT ha aperto nuove opportunità nel dominio del settore medico per quanto riguarda salvare e migliorare la qualità di vita delle persone. Ha creato nuove aree di applicazione e cambiato il modo in cui le cose vengono fatte nelle aree già esistenti come l'acquisizione dei dati, la gestione dati dei pazienti o il processo di decision making clinico.

Di seguito si procede ad elencare le principali applicazioni dell'IoT al settore sanitario. Per un elenco più completo si rimanda all'articolo [19] e [16].

1.3.1 Monitoraggio della frequenza cardiaca

Si tratta di una delle applicazioni più studiate in assoluto. La tecnica più usata per determinare le condizioni del cuore è l'ECG (electrocardiogram) ovvero il monitoraggio dell'attività elettrica del cuore attraverso l'elettrocardiogramma. L'ECG include la misurazione del semplice battito cardiaco, la determinazione del ritmo cardiaco base così come la diagnosi di aritmie, ischemie miocardiche e intervalli QT prolungati. Già svariati studi sono stati condotti sul monitoraggio IoT dell'ECG. Per esempio in [28] si effettua uno studio e una ricerca sulla possibile

implementazione di un ECG su un processore di una Raspberry Pi, permettendo di mappare il segnale attraverso un tool grafico su Internet. Lo studio cerca di far fronte alla situazione critica in India per quanto riguarda la percentuale di popolazione che soffre di malattie cardiache e l'alto costo delle macchine ECG negli ospedali. Un altro esempio può essere [8] dove si propone un'altra soluzione low cost per la trasmissione del segnale ECG. In questo caso veniva proposta un'interfaccia web accessibile contemporaneamente da più persone. Il sistema prevede la raccolta dei dati dai pazienti tramite l'elettrocardiogramma, l'invio di questi al server tramite ZigBee e la possibilità di accedervi dalle persone autorizzate per scopi di consultazione e trattamento delle malattie cardiache.

1.3.2 Rilevamento del livello di glucosio nel sangue

Il diabete è un gruppo di malattie metaboliche in cui vi sono alti livelli di glucosio nel sangue per un periodo prolungato di tempo. Il monitoraggio del glucosio rivela modelli individuali della sua variazione e aiuta nella pianificazione dei pasti e attività. In [17] è stato proposto un metodo non invasivo di rilevamento del livello di glucosio in tempo reale. In [30] invece, viene creato un sistema automatizzato di monitoraggio di glucosio e successiva iniezione di insulina in base al livello rilevato. I dati vengono rilevati da appositi sensori e mandati, tramite Wi-fi, sullo smartphone che si occuperà di processarli e decidere la necessaria dose di insulina da iniettare.

1.3.3 Monitoraggio della Temperatura corporea

Si tratta di uno dei servizi sanitari essenziali in quanto la temperatura corporea svolge un ruolo fondamentale nel mantenimento della omeostasi. Sono stati proposti vari studi sull'integrazione dell'IoT al monitoraggio della temperatura corporea come in [5] dove viene registrata e resa disponibile su mobile attraverso l'uso di alcuni sensori e di Arduino.

1.3.4 Monitoraggio della saturazione dell'ossigeno nel sangue

Il metodo più usato per il monitoraggio della saturazione dell'ossigeno del sangue è il Pulse oximetry o pulsossimetria. Il Pulse oximeter permette non solo di misurare la frequenza cardiaca ma anche di determinare l'ammontare di ossigeno presente nel corpo.

1.3.5 Monitoraggio della pressione sanguigna

La pressione sanguigna viene generalmente usata per misurare lo stato di salute di una persona. Anche in questo caso sono stati proposti dei metodi di integrazione con l'IoT per permettere una maggiore fruibilità dei dati. [35] propone un sistema che permette sia di raccogliere i dati, come il battito cardiaco e la pressione del sangue, che di inviarli ad un web server tramite connessione wireless. Inoltre la parte server metterebbe a disposizione una API RESTful e un'interfaccia grafica per permettere ai dispositivi collegati di inviare e ricevere informazioni.

1.3.6 Sistemi di riabilitazione

Poiché la medicina fisica e la riabilitazione possono migliorare e ristabilire l'abilità funzionale e la qualità della vita di quelle persone con impedimenti o disabilità fisiche esse rappresentano un ramo vitale della medicina. I sistemi riabilitativi sviluppati in ottica IoT stanno diventando il modo migliore per far fronte ai problemi legati all'invecchiamento della popolazione e alla carenza del personale sanitario qualificato. Le tecnologie basate sull'IoT possono formare una infrastruttura utile al supporto della consultazione in remoto durante il periodo di riabilitazione integrale. Un esempio di riabilitazione basata sull'IoT è il sistema robotico post trauma di [10]. In questo caso viene suggerito un immediato trattamento riabilitativo subito dopo il trauma in modo da stimolare fin da subito la guarigione. Nel caso specifico viene proposto un device per la riabilitazione del polso che sfrutta la Kinect camera per raccogliere e capire il movimento da fare del fisioterapista, per poi processare i dati raccolti e inviarli al robot tramite protocollo MQTT.

1.3.7 Ambulanza intelligente

Una risposta rapida ad una emergenza sanitaria può essere cruciale per prevenire delle perdite. Date queste circostanze, raggiungere gli ospedali velocemente è un problema serio. Pertanto ambulanze veloci ed intelligenti sono una diretta necessita nel settore sanitario. Sivaraj et al. [32] propone, in ottica IoMT, un sistema che si compone di due parti. Innanzitutto, i dispositivi medici monitorano i parametri vitali dei pazienti all'interno dell'ambulanza e inviano le informazioni all'ospedale per la predisposizione. In secondo luogo, per evitare ritardi, il sistema modifica automaticamente i semafori e libera il percorso dell'ambulanza.

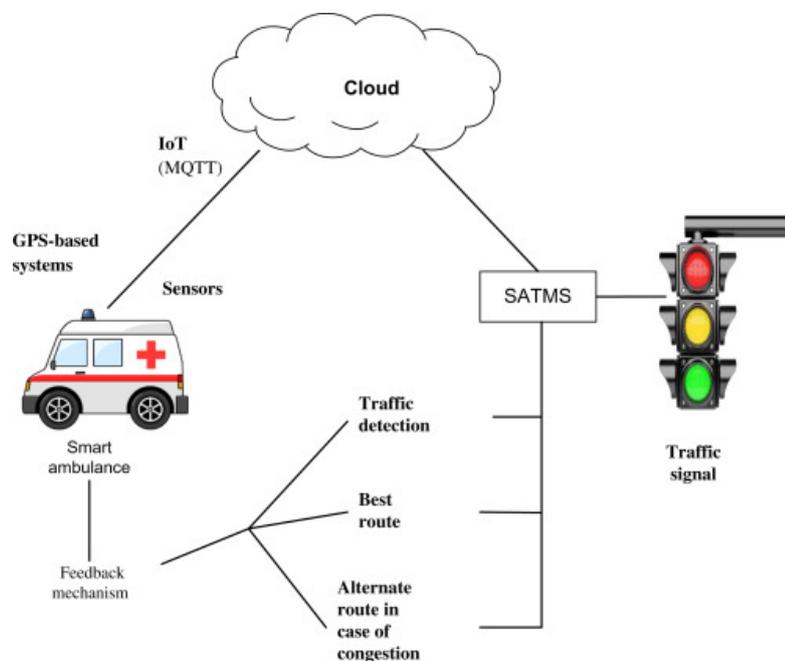


Figura 1.2: Ambulanza intelligente[11]

1.3.8 Altre applicazioni

Di seguito si elencano altre condizioni di salute per cui sono state proposte soluzioni IoT.

- **Medication Managment.** Un set completo di sensori wireless biomedici (touch, umidità e CO₂); la diagnosi e prognosi dei parametri vitali registrata dai wearable sensors; GPS, accesso al database, accesso al web, tecnologia RFID e trasmissioni multimediali.
- **Weelchair managment.** Sensori WBAN (Wireless Body Area Network, ovvero una rete che interconnette sensori indossabili con raggio molto limitato) come accelerometri o sensori che permettono di monitorare i segnali vitali della persona; comunicazioni wireless con vari nodi, capacità di percepire i dintorni.
- **Eye disorder, skin infection.** fotocamera dello smarthphone; ispezione visiva e/o pattern matching con una libreria standard di immagini.
- **Asma, fibrosi cistica, ostruzione cronica dei polmoni.** Un sistema audio integrato nello smarthphone calcola il flusso d'aria e crea grafici flusso-tempo, volume-tempo e flusso-volume.
- **Riniti e sintomi nasali.** Sempre un sistema audio integrato con microfono nello smartphone permette di riconoscere il parlato e il modo di parlare grazie ai modelli vector machine.
- **Rilevamento di melanoma.** Utilizzando la fotocamera dello smartphone è possibile effettuare il confronto tra le immagini della pelle e quelle disponibili in una libreria di immagini di pelle con melanoma.
- **Chirurgia a distanza.** Sistemi robotizzati di chirurgia, sensori per la realtà aumentata, braccia robotiche e un sistema sensoriale di feedback permetterebbero la chirurgia a distanza grazie anche alla connettività dei dati real-time.

1.3.9 Ospedale intelligente

In ultimo si effettua un piccolo approfondimento sul concetto generale di ospedale intelligente che molte volte viene citato in letteratura. Questa applicazione dell'IoMT racchiude in realtà molte delle applicazioni già dette ma le dispone in una

visione più ampia, dove ogni dispositivo, smartphone o computer è interconnesso con tutti gli altri e fa quindi parte di un'infrastruttura più grande. Con ospedale intelligente ci si riferisce all'evoluzione che sta tutt'ora avvenendo negli ospedali e più in generale in ambito sanitario in ottica IoT. Il primo passo verso un ospedale intelligente potrebbe essere semplicemente la prenotazione degli appuntamenti tramite smartphone, secondo la disponibilità del medico (già molti ospedali lo permettono in modo più o meno accessibile) oppure la visualizzazione di un referto medico o delle proprie informazioni sanitarie online. Fino ad arrivare ad un organismo perfettamente coordinato ed estremamente efficiente, dove ogni dispositivo medico (ECG, pulsossimetro, rilevatore pressione sanguigna, ecc.) mette a disposizione i propri dati poiché questi vengano analizzati dai medici, visualizzati dai pazienti o elaborati assieme ai dati di altri dispositivi.



Figura 1.3: Ospedale Intelligente[24]

1.4 Sicurezza IoMT

La sicurezza dei dispositivi IoMT è una delle sfide più importanti e significative. I dati sanitari coinvolti nei sistemi IoMT dovrebbero essere protetti in varie fasi, compresa la raccolta dei dati, trasmissione e memorizzazione. I sistemi IoMT

differiscono da altri sistemi poiché trattano dati sensibili che potrebbero influenzare la privacy dei pazienti. Inoltre, il costo medio dei dati medici è 50 volte superiore a quello dei dati sulle carte di credito, rendendoli di grande valore sul mercato nero. La sicurezza è quindi uno degli requisiti principali per il successo dei sistemi IoMT. Diversi ricercatori hanno già proposto tecniche esplicitamente progettate per sistemi IoMT e IoT, suddivisibili in crittografia simmetrica, asimmetrica e senza chiave.

1.4.1 Simmetric-Key Algorithms

La crittografia simmetrica include ogni algoritmo di crittografia basato su chiave segreta/condivisa tra due o più nodi che vogliono comunicare. La chiave deve essere generata e distribuita prima di usare il sistema asimmetrico o di iniziare la fase di comunicazione. Gli algoritmi a chiave simmetrica possono essere utilizzati per i sistemi IoMT per consentire l'accesso gerarchico ai dati del paziente e avviare connessioni sicure senza una reale impostazione. Inoltre, possono essere utilizzati anche nella fase di autenticazione a due fattori, dove essi agiscono come primo fattore, mentre altre tecniche come il riconoscimento facciale agiscono come secondo.

1.4.2 Asymmetric-Key Algorithms

Gli algoritmi a chiave asimmetrica includono algoritmi crittografici che usano due chiavi; una pubblica e una privata, dove una svolge il ruolo di validazione e cifratura mentre l'altra di firma e decifratura. Pertanto questo tipo di crittografia è nota anche come crittografia a chiave pubblica e privata. La chiave pubblica, come suggerisce il nome, è nota a chiunque, mentre la chiave privata è nota solo al suo proprietario. Alcuni degli algoritmi più conosciuti includono l'RSA, le curve ellittiche e l'algoritmo più comune e utilizzato in ambito IoMT l'ECC. Quest'ultimo è noto per la leggerezza computazionale; una chiave ECC con una grandezza di 160 bits è buona tanto quanto una chiave da 1024-bit RSA ma è 15 volte più veloce. Per citare alcuni dei più importanti algoritmi asimmetrici in ambito IoMT si può pensare alla crittografia omografica o alle firme digitali. Questi agiscono come

primo fattore nella autenticazione combinati con le smart cards (le quali vengono ampiamente usate negli ospedali) come secondo.

1.4.3 Keyless Algorithms

Gli algoritmi senza chiave permettono di assicurare un certo livello di sicurezza senza usare chiavi condivise precedentemente. Queste tecniche includono sensori per il riconoscimento biometrico, sistemi di sicurezza basati sui token o tecniche orientate ai proxy. Ricadono anche in questa categoria le tecnologie di blockchain e intelligenza artificiale. Il riconoscimento biometrico ad esempio, è una delle tecniche di sicurezza in ambito IoMT più comuni e usate. In un ambiente medico ideale, il personale medico o il paziente possono accedere alle cartelle cliniche utilizzando solo i loro dati biometrici. I fattori biometrici includono impronte digitali e sensori basati su ECG utili in caso di emergenza. I sensori di impronte digitali si basano sulla lettura dell'immagine dell'impronta digitale, mentre i sensori basati su ECG registrano le attività del battito cardiaco per crittografare i dati.

Capitolo 2

Attuali standard di interoperabilità e FHIR

Molti dei dispositivi in uso attualmente in ambiente sanitario sono generalmente interoperabili con altri dispositivi della stessa casa produttrice e possono formare tra loro una rete scalabile e controllata. Questa dipendenza, tuttavia, presenta un collo di bottiglia in quasi tutti i casi di utilizzo pratico a causa di una vasta gamma di sensori e strumenti medici. Il dispositivo di monitoraggio continuo del glucosio (CGM) di un certo fornitore, ad esempio, misura i livelli di glucosio nel sangue a intervalli regolari. Questi dati potrebbero non essere usabili in una penna per insulina (IP) di un altro fornitore [18]. É qui che nasce il problema dell'interoperabilità tra dispositivi di vendors diversi. Secondo [15] l'interoperabilità può essere definita come:

”the ability of different information and communications technology systems and software applications to communicate, to exchange data accurately, effectively, and consistently, and to use the information that has been exchanged”,

ovvero la capacità di diversi sistemi di tecnologia dell'informazione, della comunicazione e applicazioni software di comunicare, scambiare e utilizzare dati in modo accurato, efficace e coerente. Si procede quindi a descrivere i principali standard di interoperabilità.

2.1 Integrating the Healthcare Enterprise (IHE)

IHE non si definisce come uno standard ma come un'iniziativa internazionale guidata da professionisti del settore sanitario per migliorare il modo in cui i sistemi informatici in quest'ambito condividono le informazioni. IHE promuove l'uso coordinato di standard stabiliti (sia sanitari che informatici generali) per affrontare esigenze cliniche specifiche a supporto di un'assistenza ottimale al paziente. I sistemi sviluppati in conformità con IHE comunicano meglio tra loro, sono più facili da implementare e consentono agli operatori sanitari di utilizzare le informazioni in modo più efficace. IHE è organizzato e suddiviso in domini applicativi, ad esempio cardiologia, reparto dentale o la cura del paziente. Questi vengono poi analizzati e approfonditi nei Technical Framework, i quali hanno sempre un volume riguardante i profili. Ci potrebbero essere inoltre volumi riguardanti le transazioni e le appendici.

2.2 ISO/IEEE 11073

IEEE 11073 sono una famiglia di standard con l'obiettivo di permettere una comunicazione interoperabile tra dispositivi medici diversi e sistemi di computer esterni. Gli obiettivi principali di IEEE sono:

- Fornire un'interoperabilità "plug-and-play" in tempo reale per i dispositivi medici relativi ai pazienti. Plug-and-play indica che non serve l'intervento umano nel momento in cui un dispositivo viene collegato alla rete e non serve una configurazione manuale per la sua comunicazione.
- Facilitare lo scambio efficiente dei dati relativi ai pazienti e ai dispositivi medici, acquisiti nei point-of-care (POC) e in tutti gli ambienti sanitari.

Alcuni di questi standard vengono classificati come "core". In figura 2.1 è possibile vedere il numero dello standard e il suo contenuto.

La famiglia ISO/IEEE 11073 si basa su un paradigma di gestione dei sistemi orientato agli oggetti. I dati (ad esempio misurazione, stato) sono modellati sotto forma di classi, le cui istanze sono accessibili e manipolabili utilizzando un protocollo di accesso agli oggetti.

IEEE Standard	Content
11073-10101	Nomenclature
11073-10201	Domain Information Model (DIM)
11073-20101	Application Profile, Communication Model
11073-30xxx	Transport Profiles
11073-10207 (new)	Domain Information & Service Model
11073-20701 (new)	Architecture & Binding
11073-20702 (new)	Medical DPWS

Figura 2.1: Standard core[20]

2.2.1 ISO/IEEE 11073 10201

Il Domain Information Model(DIM) definisce l'insieme generale di classi, attributi, metodi e funzioni di accesso necessarie per la comunicazione dei dispositivi medici, esso viene definito in IEEE 11073-10201. Come parte della famiglia di standard 11073 il DIM soddisfa determinati requisiti[4]:

- Definisce un modello object-oriented che rappresenta le informazioni e funzioni rilevanti incontrare nel dominio delle comunicazioni tra dispositivi medici, includendo misurazioni, dati contestuali e metodi di controllo dei device.
- Fornisce specifiche dettagliate delle classi definite in questo modello orientato agli oggetti compresi attributi e metodi.
- Definisce un modello di servizio per la comunicazione dei dispositivi medici che fornisca l'accesso alle istanze dell'oggetto, i suoi attributi e i suoi metodi.
- Utilizza la nomenclatura definita in ISO/IEEE 11073-10101 (ad esempio codici per la descrizione strutturale di un dispositivo, di misurazioni, parametri, unità) per identificare tutti gli elementi nel modello.
- Essere utilizzabile per la definizione di protocolli di comunicazione dati e per la definizione formati di archiviazione di file.
- Definisce i requisiti di conformità.
- Essere estensibile ed espandibile per incorporare esigenze future nel framework di modellazione definito.

2.3 HL7

Health Level Seven è una famiglia di standard internazionali per il trasferimento e la comunicazione di dati clinici tra dispositivi medici. Il numero 7 fa riferimento al livello 7 nel modello ISO/OSI ovvero il livello di applicazione. Questi standard sono stati creati dall'organizzazione non-profit internazionale Health Level Seven International. Fondata nel 1987 è nata fin da subito con l'obiettivo di produrre degli standard che permettessero uno scambio di dati tra i dispositivi in ambito sanitario. L'organizzazione ha definito nel corso del tempo vari standard, i più importanti sono:

- HL7 Versione 2.x
- HL7 Versione 3.x
- HL7 FHIR

2.3.1 HL7 2.x

Creato nel 1989 come protocollo standard di messaggistica è stato largamente usato da molte organizzazioni sanitarie ed è tutt'ora lo standard più utilizzato. Secondo il sito ufficiale di HL7 il 95% delle organizzazioni sanitario usa HL7 v2.x. HL7 v2 è stato sviluppato con l'obiettivo di integrare sistemi disparati all'interno di un ospedale e non scambiare informazioni tra diversi ospedali. Per esempio, aveva lo scopo di colmare il divario tra il sistema di fatturazione di un ospedale e il suo sistema di laboratorio. Nonostante una serie di aggiornamenti di versioni compatibili con le versioni precedenti, v2 ha due difetti degni di nota. Innanzitutto, la v2 non offre informazioni cliniche rigorose per i modelli di dati longitudinali e i diversi tipi di dati richiesti che è un ovvio prerequisito per l'interoperabilità. Come protocollo orientato ai messaggi, v2 invece si concentra sulla formattazione del messaggio stesso e non su come sono le informazioni sono rappresentate. In secondo luogo, la v2 manca di standard completi necessari per la sua consistenza. Arrivando al vero e proprio formato dei messaggi, HL7 v2.x usa un formato di messaggio diviso in segmenti, i segmenti sono suddivisi in campi dal delimitatore '|' (viene indicato anche se il campo è vuoto). Campi che si ripetono contengono il simbolo '~' tra

loro. I campi si possono suddividere in componenti con il delimitatore '^', i quali a loro volta si possono suddividere in sub componenti con il simbolo '&'. Infine è possibile usare il carattere di escape '\ ' per scrivere caratteri speciali, per esempio per scrivere un nome con la e commerciale. I tipi di dati sono gli elementi costitutivi dei messaggi. I dati sono strutturati in modo che entrambi le parti occupate nella comunicazione o scambio li possa leggere o scrivere utilizzando la stessa sintassi. È importante che il tipo di dati supporti gli elementi richiesti. Un campo che richiede un numero non supporterà un dato che è un codice alfanumerico. Tutti i tipi di dati sono fondamentalmente stringhe di caratteri formattate. I tipi di dati primitivi sono gli elementi costitutivi di base. Essi includono stringhe e i tipi di dati numerici. I dati complessi possono essere rappresentati da tipi di dati composti, formati da un gruppo di sub-componenti. Questi sub-componenti hanno un tipo di dati a sé stante.

2.3.2 HL7 3.x

Iniziato a sviluppare a partire dal 1995 ma pubblicato come standard solo nel 2005 HL7 v3 è stato un importante precursore di FHIR ma anche un grande fallimento. HL7 v3 introduce il Reference Information Model (RIM) e si basa su un Framework di sviluppo formale (HDF) e orientato agli oggetti. Il RIM costituisce le fondamenta della versione 3, esprime il contenuto dei dati necessari in uno specifico contesto clinico e fornisce una rappresentazione esplicita di questi. Esso identifica il ciclo di vita di un messaggio o di un gruppo di messaggi correlati. È un modello condiviso tra tutti i domini, quindi quello da cui tutti i domini creano i propri messaggi. L'HDF è un processo in continua evoluzione che cerca di sviluppare specifiche che facilitino l'interoperabilità tra i sistemi sanitari. Questo framework di sviluppo descrive in dettaglio il processo metodologico di sviluppo dello standard HL7, documentando non solo la messaggistica, ma anche gli strumenti, gli attori, le regole e gli artefatti rilevanti. Il modello più utilizzato per la rappresentazione è l'Unified Modeling Language (UML) che permette una visualizzazione chiara ed efficace. Il formato dei messaggi di HL7 v3 è estremamente differente rispetto a quello di v2, questo ha portato ad una inevitabile non compatibilità tra i messaggi di v3 e v2. Uno dei fattori più incisivi nel fallimento di questo standard è stato proprio

il formato dei suoi messaggi, che risultava difficilmente leggibile e generalmente complicato. Di seguito un esempio di messaggio HL7 v3.

Listato 2.1: Esempio di messaggio semplificato di HL7 v3

```

1 <POLB_IN224200 ITSVersion="XML_1.0" xmlns="urn:hl7-org:
  v3"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <id root="2.16.840.1.113883.19.1122.7" extension="
  CNTRL-3456"/>
4   <creationTime value="200202150930-0400"/>
5   <!-- The version of the datatypes/RIM/vocabulary used
  is that of May 2006 -->
6   <versionCode code="2006-05"/>
7   <!-- interaction id= Observation Event Complete, w/o
  Receiver Responsibilities -->
8   <interactionId root="2.16.840.1.113883.1.6" extension=
  "POLB_IN224200"/>
9 </POLB_IN224200>

```

2.4 HL7 Fast Healthcare Interoperability Resources (FHIR)

La parola "Fast" nel nome si riferisce al fatto che lo standard sia stato creato per essere veloce da applicare ed usare. Il termine "Resources" si riferisce invece a come l'intero standard è suddiviso, similmente al web e alla suddivisione delle risorse come pagine web. FHIR è composto e si suddivide in risorse. È un standard aperto e gratuito per l'interoperabilità tra dispositivi medici. Esso può essere suddiviso in 5 parti principali:

- Un modello di dati robusto per descrivere i dati medici e amministrativi.
- Un'API RESTful per interagire con i dati usando JSON o XML.

2.4. HL7 FAST HEALTHCARE INTEROPERABILITY RESOURCES (FHIR)27

- Un set di strumenti open source per implementare e testare le applicazioni FHIR.
- Un insieme di Server FHIR nel mondo con cui interagire.
- Una community di persone che progettano ed implementano FHIR e che lavorano insieme.

2.4.1 Struttura e risorse

Diversamente dagli standard precedenti di HL7, FHIR è stato creato e pensato da sviluppatori per gli sviluppatori. FHIR definisce circa 150 risorse come: Patient resource, Observation resource o DiagnosticReport resource. Ogni risorsa viene definita in termini di scopi ed usi della stessa per poi passare al contenuto della risorsa che viene spiegato in termini strutturali e di file (XML, JSON, ecc.). Una risorsa è strutturata come in figura 2.2.



Figura 2.2: Struttura di una risorsa

Nella prima sezione dei Metadata troviamo l'ID della risorsa, la versione della risorsa, tag e profili. La sezione delle estensioni copre ogni informazione che non fa parte dell'80% nella regola dell'80:20, ovvero che le specifiche di FHIR coprono ciò di cui hanno bisogno almeno l'80% dei sistemi sanitari mondiali. Il resto 20% è lasciato allo sviluppatore che ha la possibilità di estendere le risorse per i propri

casi d'uso. La sezione di narrativa esplica in linguaggio naturale e leggibile da una persona come quella risorsa è strutturata e cosa contiene. Questa sezione è opzionale ma fortemente consigliata. L'ultima sezione è il vero e proprio contenuto della risorsa. Prendendo sempre come esempio il web possiamo vedere FHIR e le sue risorse come un Web of Resources, dove ogni risorsa contiene uno o più hyper-link alla pagina di un'altra risorsa creando una vera e propria rete di risorse.

2.4.2 Profili ed estensioni

Come già anticipato FHIR fornisce un set di specifiche standard per risolvere il problema della interoperabilità. Tuttavia, non sempre questa serie di blocchi standard permettono una corretta modellazione di un sistema applicativo. In questi casi serve un adattamento delle risorse anche detto Profiling. Si può creare un profilo a partire da una risorsa, da un data type, da un altro profilo o da un'estensione. FHIR definisce i Core Profile, questi potrebbero essere vincolati e ridefiniti per un uso nazionale, regionale o a livello di singole organizzazioni. Mano a mano che vengono aggiunti vincoli a questi profili, il numero di risorse con cui potranno interfacciarsi sarà sempre minore (figura esplicativa in 2.4.2).

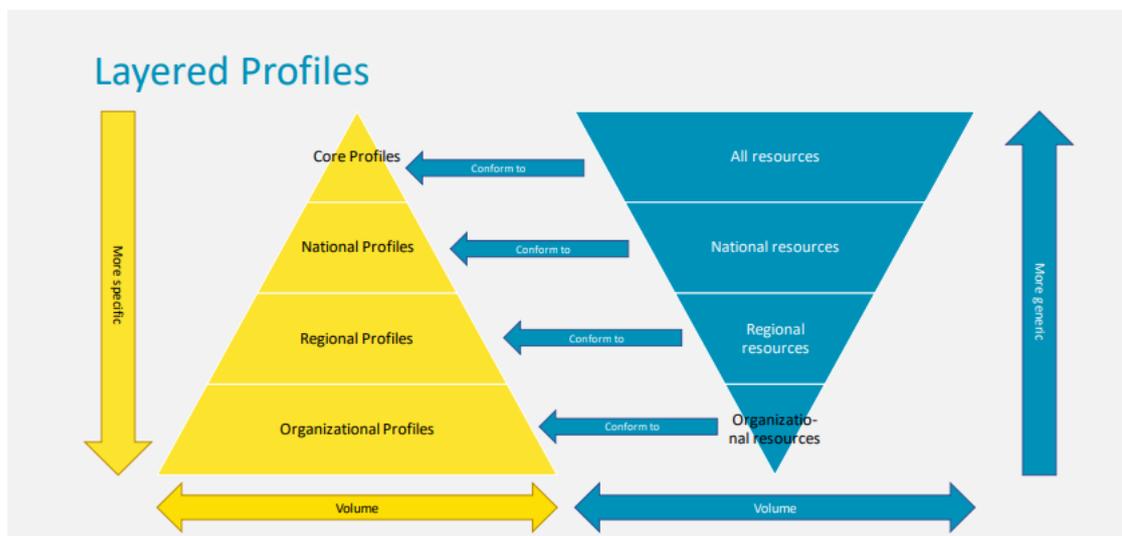


Figura 2.3: A sinistra i livelli dei profili dal core a quelli organizzativi, a destra il restringimento delle risorse conformi.

Le estensioni invece permettono di definire ed introdurre degli elementi custom a profili o risorse, come potrebbe essere l'etnia di un paziente. Una volta definita un'estensione questa potrà essere usata e riusata da tanti profili diversi. Di default FHIR permette di usare un numero illimitato di estensioni all'interno di una risorsa, tuttavia un profilo può vincolare il numero di estensioni possibili o proibirne/disabilitarne alcune.

2.4.3 API RESTful

È utile infine parlare della ricerca delle risorse e del loro uso nelle API RESTful. FHIR è descritto come una specifica "RESTful" basata sull'uso comune del termine REST. In pratica, FHIR supporta solo il livello 2 del modello REST Maturity come parte della specifica principale, sebbene la piena conformità al livello 3 sia possibile tramite l'uso di estensioni. Poiché FHIR è uno standard, si basa sulla standardizzazione delle strutture e delle interfacce delle risorse. Questo può essere considerato una violazione dei principi REST, ma è fondamentale per garantire un'interoperabilità coerente tra sistemi diversi. Ogni "tipo di risorsa" ha lo stesso insieme di interazioni definite che possono essere utilizzate per gestire le risorse in modo altamente granulare. Le applicazioni che dichiarano la conformità a questo framework dichiarano di essere conformi a "RESTful FHIR". Si noti che in questo framework RESTful, le transazioni vengono eseguite direttamente sulla risorsa del server utilizzando una richiesta/risposta HTTP. L'API non riguarda direttamente l'autenticazione e l'autorizzazione. Tutte le interazioni sono descritte per l'uso sincrono e viene definito anche un modello di utilizzo asincrono. L'API descrive le risorse FHIR come un insieme di operazioni (note come "interazioni") sulle risorse in cui le singole istanze di risorse sono gestite in raccolte in base al loro tipo. I server possono scegliere quali di queste interazioni sono rese disponibili e quali tipi di risorse supportano. I server DEVONO fornire una dichiarazione di capacità che specifichi quali interazioni e risorse sono supportate.

2.4.4 Ricerca

La ricerca di risorse è fondamentale per le meccaniche di FHIR. Le operazioni di ricerca attraversano un insieme esistente di risorse filtrando in base ai parametri forniti all'operazione di ricerca. Tutti gli url delle risorse contengono un url base che definisce dove risiede il server. È possibile specificare poi il tipo di risorsa (ad esempio Patient) e l'id richiesto (che può essere una combinazione di lettere e numeri fino a 64 caratteri). Oltre al parametro id che esiste per tutte le risorse, ogni tipo di risorsa FHIR definisce il proprio insieme di parametri di ricerca con i loro nomi, tipi e significati. Questi parametri di ricerca si trovano nella stessa pagina delle definizioni delle risorse. In generale, i parametri di ricerca che vengono esplicitati corrispondono a un singolo elemento nella risorsa, ma ciò non è obbligatorio e alcuni parametri di ricerca fanno riferimento allo stesso tipo di elemento in più posizioni o ad elementi derivati. Alcuni parametri di ricerca definiti dalle risorse sono associati a più di un percorso di una risorsa. Ciò significa che basterà anche solo uno di questi corrisponda perché la ricerca sia valida. Se un percorso corrisponde, nei risultati della ricerca viene restituita l'intera risorsa. I server non devono implementare nessuno dei parametri di ricerca standard (ad eccezione dell'id) e possono anche definire i propri parametri. Per una più approfondita documentazione si rimanda alla pagina di ricerca del sito ufficiale di FHIR [2].

2.4.5 Point-of-Care Device Implementation Guide

In ultimo si vuole analizzare l'ultima versione dell'Implementation Guide di FHIR. Si tratta della versione 0.3.0 - STU 1 Point-of-Care Device Implementation (In particolare si parla di una versione "ballot" ovvero ancora work-in-progress) basata su FHIR 4.0.1 . Essa definisce l'uso delle risorse FHIR per trasmettere misurazioni e dati dai point-of-care medical devices (PoCD) a sistemi informatici che gestiscono cartelle cliniche elettroniche, archiviazione di dati medici e sistemi di supporto decisionale clinico. Questa guida è stata sviluppata a partite dalla collaborazione tra HL7 Health Care Devices Working Group, Integrating the HEalthcare Enterprise (IHE) e Personal Connected Health Alliance (PCHAlliance) con l'iniziativa

2.4. HL7 FAST HEALTHCARE INTEROPERABILITY RESOURCES (FHIR)31

Devices on FHIR (DoF). Il gruppo DoF propone 3 usi diversi di FHIR attraverso l'Implementation Guide[3]:

- Massimo allineamento tra PoCD e Personal Health Devices (PHD)
- Allineamento con le specifiche di HL7 v2 del framework IHE Patient Care Device.
- Una mappatura appropriata del DIM e della terminologia di ISO/IEEE 11073

La guida fin da subito dichiara di non essere rivolta a quei dispositivi di cura personale riguardanti la salute a casa o il fitness, ma si rivolge al personale sanitario che tratta i dati dei point-of-care, a coloro che implementano sistemi sanitari e agli sviluppatori di gateway per i dispositivi. La guida si basa sulla nomenclatura di IEEE 11073 e sui profili sviluppati da IHE. Inoltre, come struttura per rappresentare un device si rifà al Domain Information Model di IEEE 11073 10201. Come già menzionato questa Implementation Guide si basa sui profili. Di seguito si elencano e spiegano brevemente i ruoli dei profili coinvolti nel modello gerarchico di un Point-of-Care device:

- **Medical Device System profile (MDS)**. Derivante dalla risorsa base di FHIR Device è la risorsa di livello più alto nel modello gerarchico di un Point-of-Care device e rappresenta il device nella sua interezza.
- **Virtual Medical Devices Profile (VMD)**. Si tratta di sottosistema medico nel modello gerarchico e rappresenta un dispositivo medico virtuale specifico. Più VMD possono comporre un MDS.
- **Channel Device profile**. I Channel Device permettono di raggruppare in un unico canale tante misurazioni diverse (Device Metric).
- **Device Metric profiles**. Indica una misurazione, calcolo, caratteristica o capacità tecnica. Tuttavia non sono necessariamente singole misurazioni quantitative ma possono anche essere enumerazioni (indicano delle qualità o delle categorie) o un insieme di valori correlati.

2.5 openEHR

openEHR è uno standard aperto nel settore informatico sanitario costituito da modelli clinici e software che vengono utilizzati per cercare e creare soluzioni al problema dell'interoperabilità tra dispositivi medici. Tutti i modelli di openEHR sono prodotti dalla comunità openEHR e gestiti da openEHR International, un'organizzazione internazionale senza scopo di lucro fondata nel 2003. In openEHR tutti i dati relativi alla salute di un individuo sono memorizzati in un Electronic Health Record (EHR) ovvero nella cartella clinica elettronica.

2.5.1 Obiettivi openEHR

Gli obiettivi di openEHR sono:

- promuovere e pubblicare le specifiche formali e i requisiti per la rappresentazione e la comunicazione di EHR.
- promuovere e pubblicare l'architettura e i modelli delle informazioni salvate negli EHR.
- lavorare a distanza ravvicinata con il corpo degli altri standard, in modo che le specifiche e il design di openEHR incorpori gli aspetti migliori degli standard più rilevanti, e che sia soggetto di buoni principi e pratiche di design.
- validare l'architettura degli EHR attraverso una implementazione integrazione e una valutazione clinica.
- mantenere un'implementazione delle referenze open source per migliorare la scelta di tools disponibili al supporto dei sistemi clinici.
- collaborare con altri gruppi di lavoro con l'obiettivo di realizzare un sistema sanitario interoperabile, basato sui requisiti e di alta qualità.

2.5.2 Applicazione openEHR COVID-19

La diffusione del COVID-19 ha messo a dura a prova gli ospedali di tutto il mondo e dato vita a nuove sfide per i sistemi informatici in ambito medico. In [22] vie-

ne effettuato uno studio sullo sviluppo di un modello di dati condiviso basato su openEHR per migliorare l'interoperabilità i sistemi che si occupano della diagnosi e del trattamento del COVID-19. In particolare viene scelta, come fonte di informazione e linea guida riguardo la modellazione del modello, l'ultima linea guida sul trattamento e la diagnosi del COVID-19 in China. Questa viene analizzata e vengono definiti ed estratti i dati relativi alla diagnosi e al trattamento. I dati vengono poi organizzati in domini e mappe mentali. In ultimo è stata effettuata una ricerca sui possibili archetipi esistenti che potessero rappresentare i concetti, creandone di nuovi se necessario. Nell'articolo è stato scelto di utilizzare openEHR come standard tra quelli esistenti in quanto servisse un modello di dati dinamico, flessibile e che potesse stare al passo con l'evoluzione dei requisiti riguardanti il COVID-19.

Capitolo 3

Emogasanalisi e caso di studio

3.1 Cos'è

L'esame di emogasanalisi permette di capire esattamente le quantità di gas presenti nel sangue, in particolare di ossigeno e anidride carbonica, e l'acidità del sangue stesso (pH). Le misurazioni dei gas presenti nel sangue permettono di capire se lo scambio gassoso a livello degli alveoli polmonari sta avvenendo con efficacia. Questa tecnica viene usata per rilevare diverse malattie polmonari e respiratorie, malattie del sonno, insufficienze renali o cardiache e infezioni. L'emogasanalisi consiste nell'analisi di una provetta di sangue prelevata da un paziente. L'emogasanalisi permette anche di capire se vi è una carenza di basi nel sangue, inoltre quantifica i livelli degli elettroliti, ovvero sodio, potassio, calcio e cloro (particolarmente importante dopo il processo di dialisi).

3.2 Dispositivi EGA

É in questo scenario che entrano in gioco gli apparecchi di emogasanalisi o emogasanalizzatori. Essi permettono di analizzare praticamente in tempo reale un prelievo di sangue e in poche decine di minuti fornire una lista dettagliata delle quantità descritte sopra. Vi sono una varietà di questi apparecchi in commercio venduti da diverse case di produzione. Tra i più famosi troviamo l'ABL800 FLEX della Radiometer o il PREMIER 5000 della GEM entrambi con schermo touch

e possibilità di scambiare messaggi in formato HL7. Di particolare importanza è stato il primo in quanto pioniere dell'automazione delle misure. Dal prelievo del sangue al referto inviato ad un PC o monitor la prima versione dell'ABL800 permetteva un processo sicuro per l'operatore sanitario e immune da errori preanalitici, di identificazione del paziente e di trascrizione. Il salto in avanti è avvenuto, sempre da parte della radiometer, con l'ABL 90 che oltre ad integrare un' alta connettività permette di essere portatile essendo previsto di una batteria per l'alimentazione. Per quanto riguarda i protocolli di messaggistica la Radiometer è dal 1998 che implementa sui suoi dispositivi il protocollo HL7[25].

3.2.1 RAPIDPoint 500 Siemens

Entrando nello specifico del caso di studio, all'ospedale M. Bufalini di Cesena viene utilizzato il dispositivo RapidPoint 500 della Siemens. Si tratta di un



Figura 3.1: Dispositivo EGA RAPIDPoint 500

sistema a cartuccia completamente automatico. Si distingue dagli altri dispositivi per la sua facilità d'uso e per la totale assenza di manutenzione. Consente di rilevare automaticamente la presenza di bolle d'aria o di coaguli nel sangue e gestirli autonomamente in modo da non sprecare una cartuccia. Impiega una varietà di

tecniche per la misurazione di determinate sostanze attraverso determinati sensori, in figura 3.2 vengono elencati i sensori con le tecniche.

Sensore	Tecnica di misura
pH, Na ⁺ , K ⁺ , Ca ⁺⁺ , Cl ⁻	metodo potenziometrico che fa uso della tecnologia standard ad elettrodi ione- selettivi ¹ (ISE)
Riferimento	elettrodo in argento/argento in cloruro di potassio e cloruro d'argento
PCO ₂	metodo potenziometrico modificato, basato sui principi dell'elettrodo di Severinghaus ²
PO ₂	metodo amperometrico basato sui principi dell'elettrodo di Clark ³
Glucosio	metodo amperometrico che fa uso di un elettrodo enzimatico contenente glucosio-ossidasi ³
Lattato	metodo amperometrico che fa uso di un elettrodo enzimatico contenente lattato-ossidasi ³
Emoglobina	Metodo spettrofotometrico
Bilirubina	Metodo spettrofotometrico

Figura 3.2: Sensori con tecniche di misurazione

Dopo aver analizzato i dati, essi vengono visualizzati sull'apposito schermo del dispositivo. Il dispositivo ha chiaramente una memoria limitata, tuttavia attraverso un periodico backup dei dati su chiavetta USB è possibile "estendere" questa memoria illimitatamente. I dati su chiavetta USB potranno poi essere letti da qualsiasi PC. Il referto del campione viene stampato automaticamente a fine analisi, anche se può essere ristampato in qualsiasi momento. Il dispositivo, operante su piattaforma Windows XP, permette di essere collegato ad un sistema di gestione RapidComm (di proprietà della Siemens) tramite rete Ethernet, quindi oltre a stampare i dati questi possono essere inviati sulla rete. Il programma RAPIDComm utilizza la versione 2.4 dello standard HL7 per formattare i messaggi in uscita e per elaborare i messaggi dai sistemi ospedalieri. Il programma può anche formattare i messaggi in uscita ed elaborare i messaggi in entrata compatibili con HL7 versione 2.3, utilizzati in alcune strutture. Questa modifica può essere ottenuta mediante un'impostazione in ciascuna delle interfacce di comunicazione. Sebbene il sistema RAPIDComm possa funzionare con sistemi che producono file formattati in 2.1 e 2.2, etichetterà ciascuno dei messaggi che produce come 2.3 o

2.4. Le informazioni inviate utilizzando lo standard HL7 sono strutturate in singoli messaggi. Ogni tipo di messaggio HL7 ha uno scopo e una struttura specifici. RAPIDComm utilizza i seguenti segmenti per inviare e ricevere dati: MSH, PID, PV1, ORC, OBR, OBX, NTE, e MSA. I dati di altri segmenti vengono ignorati. Tutte le comunicazioni tra RAPIDComm e i sistemi ospedalieri utilizzano TCP/IP.

3.3 Analisi dominio e risorse

Dapprima si cerca di inquadrare gli attori coinvolti nel nostro ambito applicativo prendendo il caso nel particolare e come dominio quello dell'ospedale. Gli attori principali sono i Dispositivi (nel caso specifico l'EGA), i pazienti e le Misurazioni o Osservazioni. Le misurazioni vengono raccolte in un referto e rappresentano tutti i possibili valori letti dai dispositivi EGA (pH, CO₂, O₂, sodio, potassio, ecc.). Come modello di dominio generico possiamo immaginarlo come in figura 3.3. Entrando nel contesto di FHIR tuttavia, esso ci propone delle risorse ben specifiche

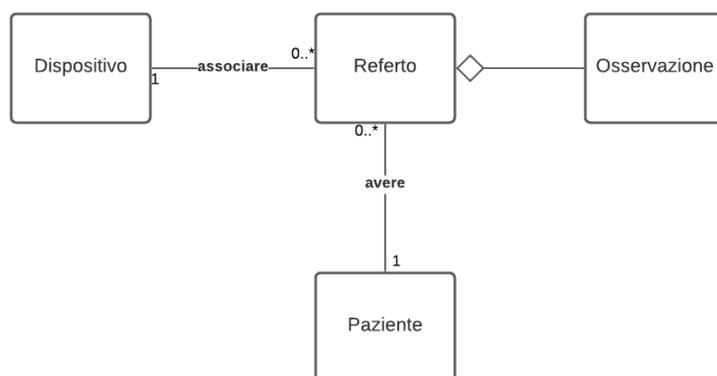


Figura 3.3: Diagramma generico del dominio

per modellare questo tipo di dominio. Troviamo infatti delle corrispondenze dirette con alcune delle classi descritte, in particolare la classe Device per i dispositivi, la classe Observation per le misurazioni e la classe Patient per i pazienti. Il referto può essere visto come un DiagnosticReport in FHIR. Viene messa a disposizione inoltre una risorsa molto importante, il DeviceMetric che ci permette di descrivere in modo dinamico alcuni importanti aspetti del dispositivo. Il DeviceMetric è

referenziato dall'Observation e contiene un riferimento al Device. Si analizzano le risorse di FHIR principali per il nostro dominio.

3.3.1 Device

La risorsa Device descrive in modo statico un certo dispositivo medico o non medico. Si tratta di una risorsa amministrativa che permette di tracciare le istanze individuali di un dispositivo e le loro posizioni. Viene referenziato da altre risorse per capire quale device ha effettuato una certa azione come la Procedure o la Observation. I Device hanno una varietà di attributi come l'identificatore, la definizione, la data di produzione del dispositivo, o la posizione attuale. Inoltre essi possono contenere un riferimento al DeviceName (con il nome del dispositivo) e alla versione.

3.3.2 DeviceMetric

La risorsa DeviceMetric descrive un calcolo, misurazione o impostazione di un dispositivo medico. Essa descrive le proprietà statiche del dispositivo ma può essere utilizzata anche per alcune proprietà dinamica estremamente importanti come l'ora, il tipo e le modalità dell'ultima misurazione e lo stato della metrica. Quest'ultimo viene descritto dal campo operationalStatus che può assumere i valori 'on', 'off', 'standby', 'entered-in-error', lo stato ci permette di tenere traccia in modo dinamico delle sottoparti che compongono un dispositivo, quindi di sapere quali sono attive e quali no.

3.3.3 Observation e DiagnosticReport

Le Observation sono l'elemento centrale di FHIR e dell'ambito healthcare, possono contenere un campo device con un riferimento al Device o al DeviceMetric e un campo subject con un riferimento al Patient, Group, Device o Location. Esse rappresentano il più delle volte una coppia nome valore che indica una certa misurazione. Le Observation possono essere utilizzate per i valori dei segni vitali come la pressione sanguigna, il peso corporeo o la temperatura. La risorsa DiagnosticReport fornisce un contesto clinico o del flusso di lavoro per una serie

di osservazioni, essa contiene un riferimento alle osservazioni che la compongono. Può contenere immagini, commenti testuali e codici riguardanti la procedura diagnostica conclusa.

3.3.4 Diagramma risorse FHIR del dominio

Oltre alle risorse già elencate troviamo nel diagramma la risorsa Patient che, come indica il nome, contiene le informazioni amministrative riguardanti un paziente (quindi un individuo sotto cure mediche) tra cui l'id, un flag booleano active che indica se il record del paziente è ancora attivo, oltre a tutti gli attributi anagrafici. Viene riportato nello schema in quanto viene referenziato da DiagnosticReport. Si è deciso di inserire questo riferimento e non quello a Device in quanto è possibile comunque risalire al Device che ha prodotto quel report attraverso le Observation (sempre referenziate da DiagnosticReport). 3.3.

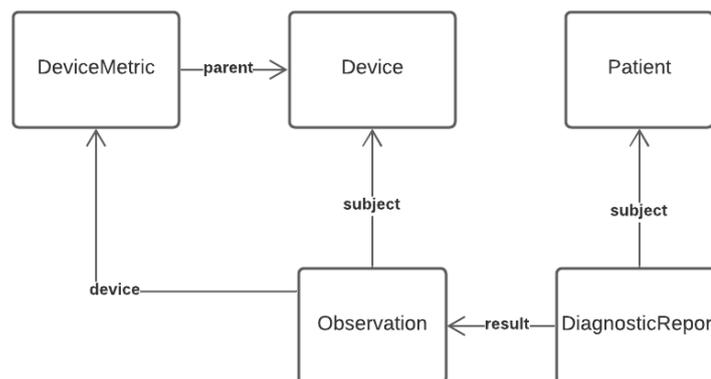


Figura 3.4: Diagramma FHIR del dominio

3.4 Modellazione Risorse PoCD Implementation Guide

In questa sezione si cercherà di modellare le risorse FHIR in un contesto reale come quello in esame. L'obiettivo è rappresentare concetti o elementi reali tramite

le risorse definite da FHIR seguendo la PoCD Implementation Guide. Le risorse che partecipano in questa modellazione sono: il Medical Device System, il Virtual Medical Device, il Channel, il DeviceMetric e l'Observation (senza contare i riferimenti delle Observation ai Patient e del MDS alla Location). Si segue quindi la struttura descritta dalla guida dove alla base troviamo le Observation che contengono un riferimento alle corrispondenti DeviceMetric, queste vengono raggruppate in Channel i quali hanno un riferimento al VMD che a sua volta si riferisce ad un MDS.

3.4.1 MDS e VMD, top level Device

In cima alla gerarchia troviamo il Medical Device System e il Virtual Medical Devices. Il Medical Device System rappresenta il dispositivo (in questo caso l'EGA) nel suo complesso, il PoCD definisce che l'identifier, il type e il campo udiCarrier (codice a barre) dovrebbero essere presenti. Viene specificato che il campo identifier dovrebbe includere una codifica esadecimale e che il tipo dovrebbe essere codificato secondo la nomenclatura ISO/IEEE 11073. Il VMD dovrebbe presentare un campo parent che si riferisce alla risorsa top-level MDS. È stato deciso di inserire un solo VMD con cardinalità 0 e 1, in questo modo è possibile registrare un generico dispositivo di emogasanalisi (figura 3.5).

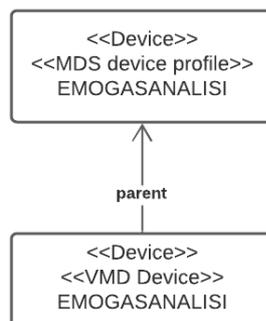


Figura 3.5: MDS e VMD

3.4.2 Channel

I valori letti dal dispositivo EGA si suddividono in 5 categorie: acido-base, co-ossimetria, ossigenazione, elettroliti e metaboliti (figura 3.6).

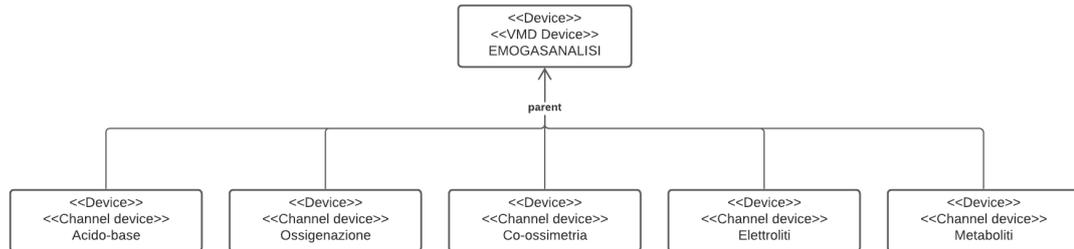


Figura 3.6: Channels

3.4.3 DeviceMetric e Observation

Di seguito si rappresentano tutte le DeviceMetric relative ai singoli channel facendo riferimento ai valori letti dal RAPIDPoint 500. Non si esplicitano per ogni DeviceMetric la relativa Observation che è legata da una cardinalità 1 a 1.

- **Acido-base.** Il channel Acido-base fa riferimento all'equilibrio acido base e raggruppa le misurazioni in figura 3.7. Tra queste troviamo il pH, la pressione parziale di ossigeno PO2 e la pressione parziale di anidride carbonica PCO2.
- **Co-ossimetro.** Il co-ossimetro è in grado di misurare la saturazione dell'ossigeno dell'emoglobina e di distinguere la presenza di emoglobina legata al monossido di carbonio. Diagramma delle classi del channel co-ossimetro in figura 3.8
- **Elettroliti.** Gli elettroliti sono minerali che si trovano nei liquidi del corpo (urine, sangue, ecc.) sotto forma di ioni positivi (cationi) e negativi (anioni), come sodio, potassio, calcio, bicarbonati e fosfati. Il dispositivo EGA RAPIDPoint500 fornisce i valori per i cationi sodio, calcio e potassio, per l'anione cloro e per il gap anionico AnGap (figura 3.9).

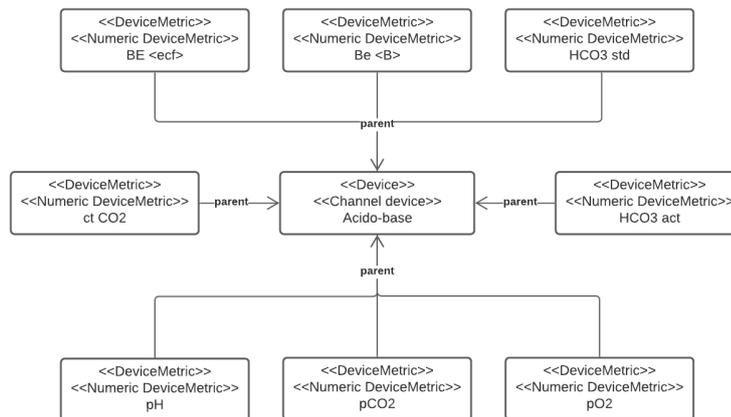


Figura 3.7: Acido-base

- **Metaboliti.** I metaboliti sono un prodotto finale o intermedio del metabolismo. Nel nostro caso troviamo il glucosio e il lattato (figura 3.10).
- **Ossigenazione.** In ultimo due valori riguardanti l'ossigeno ovvero il contenuto di ossigeno ctO2 e la capacità di ossigeno dell'emoglobina BO2 (figura 3.11).
- **Numeric Observation.** Per l'observation si fornisce un solo esempio in si trova sempre in rapporto 1 a 1 con la DeviceMetric corrispondente (figura 3.12). Da notare che il riferimento che si trova dentro ad Observation alla sua Metric si chiama 'device' e non 'parent'. Inoltre definisce un 'subject', che dovrebbe essere presente e riferirsi ad un paziente o ad un MDS device, un 'interpretation', 'code', 'valueQuantity' e 'valueRatio' (nel caso non ci sia un valore deve essere presente 'dataAbsentReason').

3.4.4 Applicazione del modello

In questo capitolo si è partiti come una descrizione del dispositivo in esame, per poi proseguire con una possibile modellazione del dispositivo di emogasanalisi seguendo la Point of Care Device Implementation Guide e le risorse FHIR messe a disposizione. Nel prossimo capitolo, dopo aver introdotto il progetto esistente

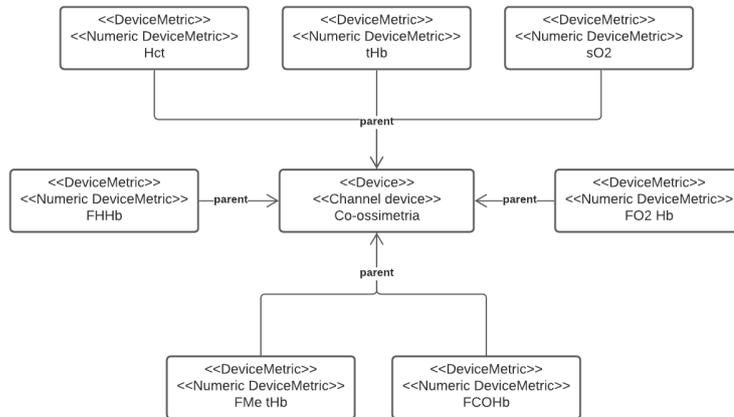


Figura 3.8: Co-ossimetria

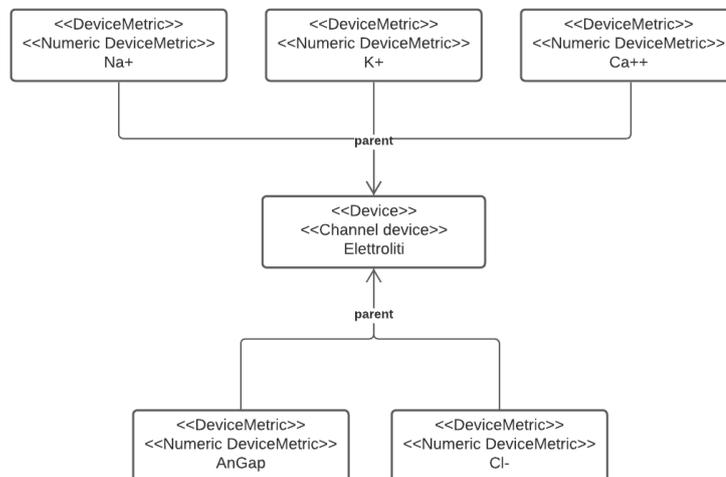


Figura 3.9: Elettroliti

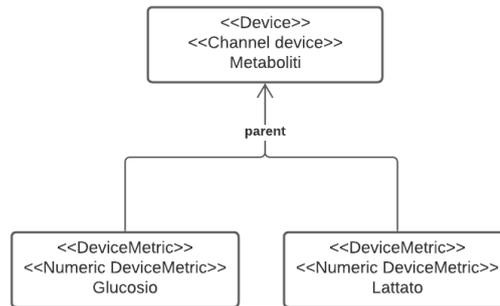


Figura 3.10: Metaboliti

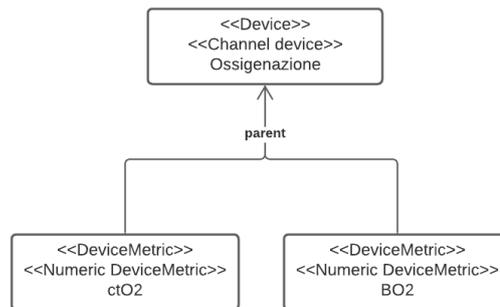


Figura 3.11: Ossigenazione

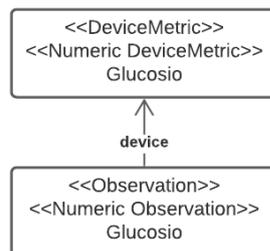


Figura 3.12: Esempio Observation

Tracking for Care, verrà analizzato come sia possibile disporre praticamente di queste risorse e di come esse possano essere salvate su un FHIR server e delle API che questo espone.

Capitolo 4

Progettazione Componente FHIR per il progetto T4C

In questo capitolo si vuole procedere dapprima con una breve illustrazione dello stato attuale del sistema e di come il progetto T4C si inserisca in esso. Si proseguirà poi con una descrizione delle tecnologie utilizzate e della progettazione di un componente del progetto T4C che sia in grado di trasformare i dati in risorse FHIR, salvare tali risorse su un server e renderle disponibili tramite chiamate API.

4.1 Tracking for Care

Il progetto già esistente di Tracking for Care(T4C) si inserisce nel contesto delle applicazioni a supporto delle attività del personale medico e si occupa di effettuare il tracciamento e la documentazione delle emergenze tempo-dipendenti(Trauma, ictus, infarto). Prendendo nello specifico il progetto Tracking for Care relativo al dispositivo EGA si descrive come esso si inserisca nell'intero sistema: Il dispositivo fisico EGA o emogasanalizzatore permette di effettuare delle analisi su un campione di sangue e tramite vari sensori ottenere delle quantità precise dei gas presenti, subito dopo questi dati vengono inviati al Gateway EGA in formato HL7 v2.4. Il software proprietario della Siemens, tramite l'applicativo RAPIDComm, riesce a leggere i messaggi e li salva sul server, questi vengono poi inviati al software T4C ega-service che si occupa di leggere i dati, salvarli su un database di tipo MongoDB

ed esporli tramite API REST. Parte fondamentale del progetto è l'EgaService che si occupa in particolare di esporre delle API REST relative agli URL delle risorse e predisporre alla creazione del server per la ricezione dei messaggi HL7. Vengono messe a disposizione tre possibili API attualmente:

- Specificando

`/egaservice/info`

viene semplicemente fornito lo stato del server che è impostato su active.

- `/egaservice/reports`

permette di ottenere tutti i reports.

- `/egaservice/reports/:reportId`

permette di ottenere un report specifico a partire dal parametro reportId.

Per la creazione di questo web service viene utilizzato il tool software Vert.x che permette di effettuare il deploy di uno o più componenti chiamati Verticle, i quali sono specializzati nello svolgere un determinato compito. Questi Verticle vengono eseguiti concorrentemente e possono comunicare tramite l'Event bus. Il sistema viene rappresentato in figura 4.1.

4.2 Integrazione con il componente proposto

Viene proposto un componente da inserire come integrazione di T4C che permetta di effettuare un parsing dei dati in risorse FHIR e di salvare queste ultime in formato Json all'interno di un server FHIR. Il server FHIR esporrà le risorse tramite chiamate API seguendo le specifiche di FHIR descritte precedentemente. Il software integrante chiamato EgaComponent si inserisce quindi tra il sistema T4C e le richieste del client e mira a soddisfare queste ultime con risorse FHIR per un sistema interoperabile.

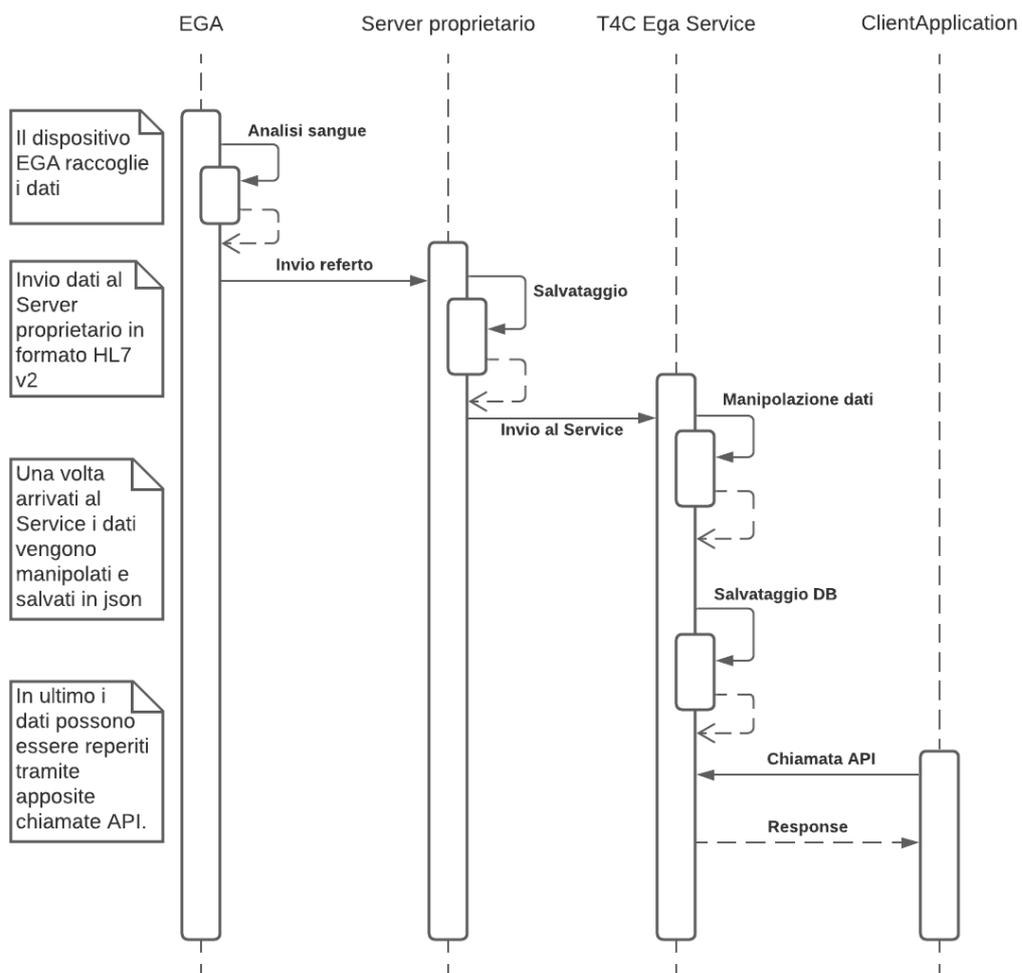


Figura 4.1: Diagramma di sequenza stato attuale

4.2.1 FHIR Server

È stato scelto di utilizzare un server FHIR in quanto esso offre già le strutture necessarie per il salvataggio delle risorse e per il loro reperimento in un formato conforme allo standard HL7 FHIR. Il server permette di ricevere vari tipi di chiamate http tra cui la Post per aggiungere una risorsa e la Get per ottenerne una. Queste vengono indicizzate rispetto al tipo di risorsa, ad esempio tutti i dispositivi si troveranno sotto "Device" mentre i report sotto "DiagnosticReport". Esistono vari FHIR server, ad esempio quello di IBM, Oracle, o firely. Ognuno di essi salva le risorse in un certo database, ad esempio Oracle FHIR server usa Oracle Database.

4.2.2 Requisiti del componente

Il componente precedentemente descritto dovrà soddisfare i seguenti **requisiti funzionali**:

- Gestire i messaggi inviati dal servizio T4C service
- Modellazione di questi messaggi in risorse FHIR
- Rappresentazione risorse in formato Json
- Connessione al server FHIR e salvataggio risorse
- Esposizione risorse tramite API specifica o chiamate al server

Tra i requisiti funzionali troviamo invece l'efficienza, ovvero il nostro software dovrà gestire le risorse in modo ottimale ed essere veloce nell'esecuzione delle operazioni richieste. Il sistema dovrà assicurare la persistenza dei report salvati tramite la creazione delle risorse e il salvataggio sull' IBM FHIR Server. In ultimo dovrà essere interoperabile, ovvero dovrà permettere ad applicativi diversi di richiedere i dati salvati.

4.2.3 T4C e Componente

Dopo aver descritto il progetto tracking for care, in particolare il T4C EgaService specifico per il dispositivo di emogasanalisi, aver presentato il componente

e i suoi requisiti e il FHIR Server si procede mostrando una visione di insieme da dispositivo fisico ai Client che richiedono le risorse, passando per il Gateway proprietario Siemens, per il T4C EgaService e per il nostro componente. In figura 4.2 il diagramma di sequenza completo.

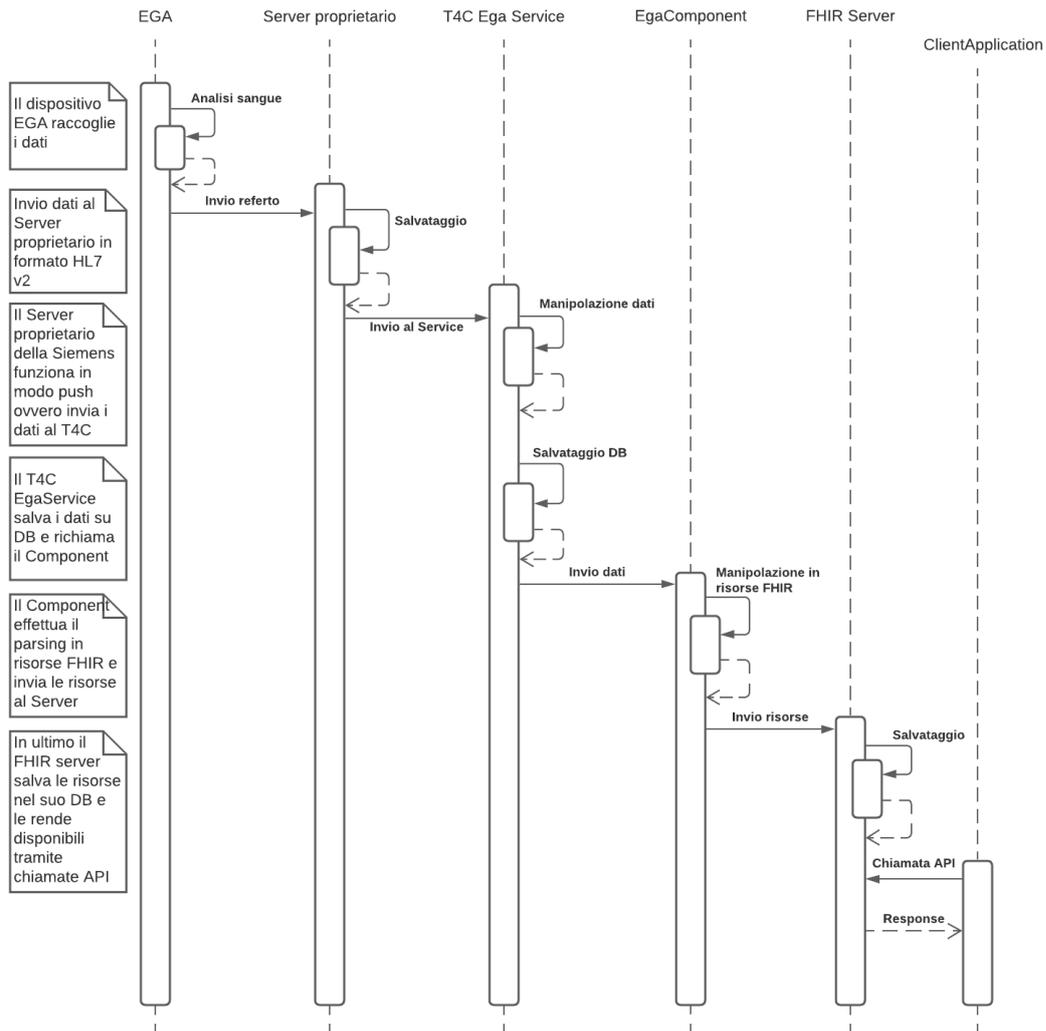


Figura 4.2: Diagramma di sequenza con EgaComponent

4.3 Progettazione

In questa sezione si entrerà nel merito di come si intende sviluppare il sistema. Per quanto riguarda il requisito riguardante la creazione di risorse FHIR, si intende interfacciarsi con il software esistente Tracking for Care che già riceve i messaggi HL7. In figura 4.3 si mostra come il componente dovrebbe interagire con il servizio T4C e sfruttare il parser per la traduzione dei messaggi HL7.

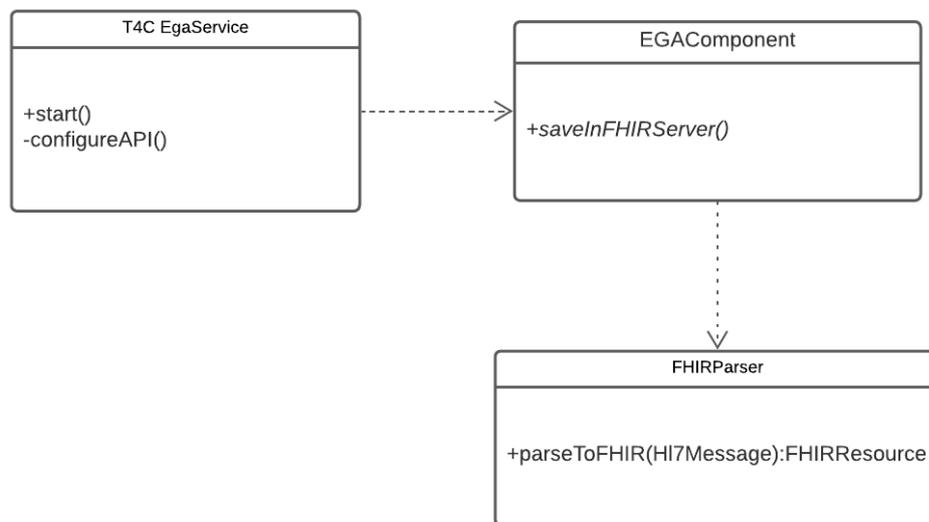


Figura 4.3: Progettazione ideale

Il Tracking for Care EgaService una volta avviato resta in attesa di messaggi da parte del gateway proprietario, li elabora e salva nel proprio database. A questo punto sfrutta il metodo `saveInFHIRServer` esposto dal nostro componente per salvare i dati nel FHIR Server. L'EGAComponent userà il FHIRParser per effettuare il parsing dei dati in risorse FHIR.

Per quanto riguarda le risorse FHIR rappresentate in java si crea un'interfaccia `FHIRResource` che descrive in generale tutte le risorse ed espone il metodo `toJson()` per convertire la risorsa specifica in formato Json. L'interfaccia può essere implementata da tutte le risorse FHIR, nel caso in esame sono state modellate le

tre risorse principali nel sistema EGA, ovvero Report (DiagnosticReport), Observation che compongono un Report e Device. Si fa riferimento all'immagine 4.4. Infine si presenta il diagramma finale dove viene rappresentato anche il FHIR-

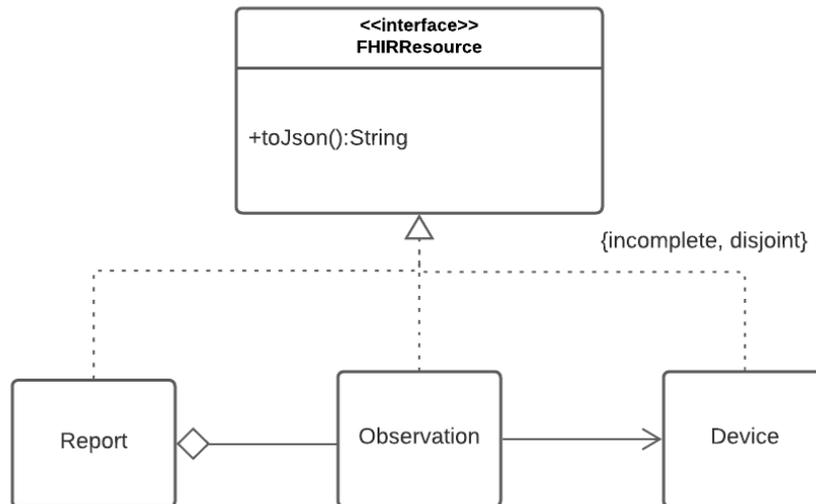


Figura 4.4: Risorse FHIR

Client che si collega al server FHIR ed espone il metodo `create()` che accetta una `FHIRResource` e permette di registrarla sul server connesso. L'`EGAComponent` si occupa quindi di utilizzare sia il `FHIRParser` per il parsing dei messaggi sia il `FHIRClient` per il salvataggio delle risorse.

4.4 Sviluppo e Testing

In questo capitolo si effettua inizialmente una delineazione delle tecnologie utilizzate per poi passare all'effettiva parte di sviluppo del componente proposto e ad un suo testing. Si vuole quindi verificare che il componente riesca effettivamente a fare ciò che è stato descritto, con particolare attenzione al salvataggio delle risorse `DiagnosticReport` nel FHIR server.

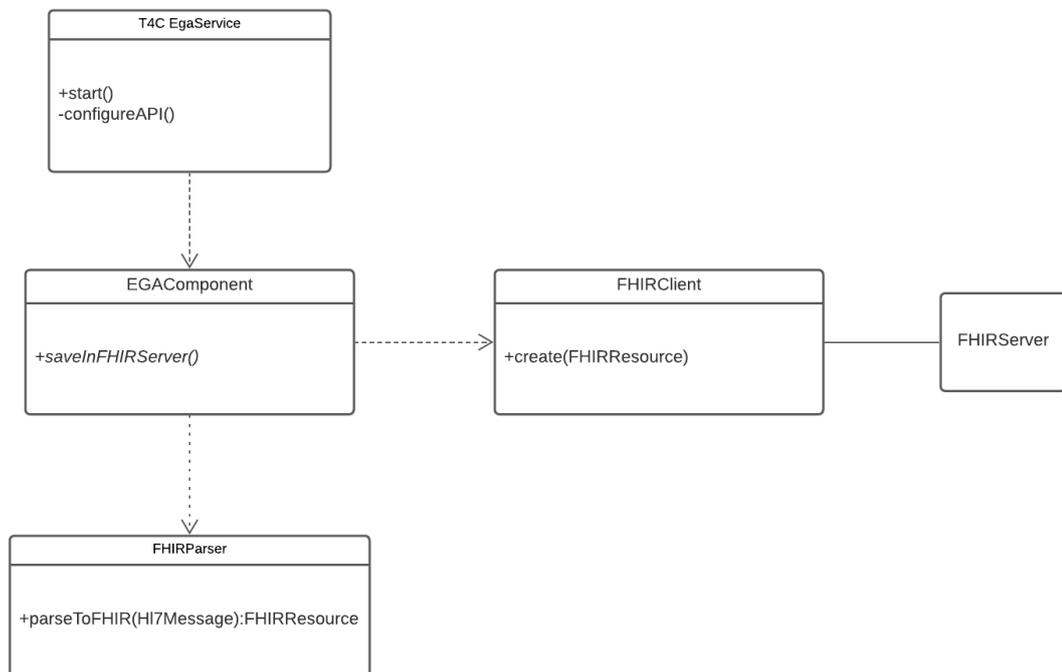


Figura 4.5: Sistema complessivo

4.4.1 Tecnologie utilizzate

È stato scelto di utilizzare il linguaggio Java per lo sviluppo del componente in quanto estremamente versatile e perfettamente adatto per lo scopo. Inoltre esso si allinea con la parte già sviluppata del T4C EgaService. Java è un linguaggio di programmazione ad alto livello a tipizzazione statica ed orientato agli oggetti. Progettato per essere indipendente dall'hardware su cui va in esecuzione deve essere necessariamente interpretato da una Java Virtual Machine, questo è sì un pregio ma anche un difetto nel momento in cui si mira ad ottenere le massime prestazioni. Si elencano e descrivono brevemente le altre tecnologie utilizzate:

- **HAPI FHIR.** Per la creazione delle risorse a partire dai dati inviati è stato scelto di utilizzare la libreria HAPI FHIR sviluppata principalmente da Smile CDR che si occupa specificatamente di scambio di dati e tecnologie in ambito healthcare ma a cui hanno contribuito persone di molte altre organizzazioni [1]. Si tratta infatti di una libreria open source che implementa in modo completo lo standard HL7 FHIR. Parte fondamentale della libreria è rappresentato dal FhirContext, esso è il punto di partenza per l'utilizzo di HAPI e funge da factory per la maggior parte delle altre API. La libreria offre inoltre un supporto completo per tutte le risorse dello standard FHIR oltre ad implementare un parser e serializer per la conversione delle risorse in formato FHIR JSON e XML.
- **IBM FHIR Server.** Come FHIR server in cui utilizzare le risorse è risultato estremamente versatile ed efficiente il server di IBM. Grazie anche al supporto del Docker è stato semplice ed intuitivo l'utilizzo dell'immagine del server in locale.
- **Docker.** Docker è un insieme di prodotti Platform as a Service (PaaS) che utilizzano la virtualizzazione a livello di sistema operativo per fornire software in pacchetti chiamati container. Il termine Docker viene in realtà usato anche per il nome dell'azienda Docker Inc. I container sono isolati l'uno dall'altro e raggruppano il software, librerie e file di configurazione; possono comunicare tra loro attraverso canali ben definiti. Poiché tutti i container condividono i servizi di un singolo kernel del sistema operativo,

utilizzano meno risorse rispetto alle macchine virtuali. L'isolamento e la sicurezza consentono di eseguire molti container contemporaneamente su un determinato host. I container sono leggeri e contengono tutto il necessario per eseguire l'applicazione, quindi non è necessario fare affidamento su ciò che è attualmente installato sull'host.

- **Postman.** Postman è una piattaforma per la creazione e l'utilizzo di API. In particolare permette di testare in modo pratico ed efficace tramite una intuitiva interfaccia grafica le API di uno specifico servizio. Nel caso in esame è stato utilizzato per verificare la corretta creazione delle risorse FHIR nel Server IBM.

4.4.2 Testing

In questa sezione si proseguirà testando il corretto funzionamento del componente proposto, con particolare riguardo ai requisiti esposti precedentemente. Come già spiegato il componente dovrebbe interfacciarsi direttamente con il Tracking for Care EGAService esponendo il metodo saveEGAResult, il quale prende in ingresso il messaggio in formato HL7. Tuttavia date le complicazioni di un testing diretto sul sistema attualmente in uso si è deciso di creare una classe HL7MessageGenerator che permette di creare messaggi realistici di un dispositivo EGA in formato HL7 v 2.4. Il generatore permette quindi di alimentare il componente creando l'oggetto Message. Una volta creato l'oggetto, questo viene passato all'EGAComponent che richiama il FHIRParser per tradurre il messaggio da HL7 a FHIR e salvato nel FHIR Server creando l'istanza del FHIRClient. Per testare l'IBM FHIR Server è stato utilizzato Postman, tramite infatti la chiamata sui DiagnosticReport è possibile verificare l'aggiunta del Report.

Listato 4.1: Esempio di DiagnosticReport aggiunto al FHIR Server

```

1 |
2 | "resourceType": "DiagnosticReport",
3 | "id": "17bf3f684a3-818f6862-8d08-4ba9-b3c1-52c4e5397a09",
4 | "meta": {

```

```
5     "versionId": "1",
6     "lastUpdated": "2021-09-17T13:33:44.483175Z"
7 },
8 "contained": [
9     {
10        "resourceType": "Observation",
11        "id": "1",
12        "status": "preliminary",
13        "code": {
14            "coding": [
15                {
16                    "system": "http://loinc.org",
17                    "code": "pH"
18                }
19            ]
20        },
21        "subject": {
22            "reference": "Device/17bcab9ed83-f7bc39cb-
23                d421-4abd-9258-8d33b60ccec3"
24        },
25        "valueQuantity": {
26            "value": 7.5,
27            "system": "http://unitsofmeasure.org"
28        }
29    },
30    ...]
```

Nel listato 4.1 si può vedere l'esempio di un DiagnosticReport aggiunto tramite il componente, il file è stato chiaramente tagliato date le innumerevoli Observation presenti. Una volta caricate le risorse sul FHIR Server è possibile effettuare delle chiamate Http sulle API esposte dal Server. Di seguito si elencano e descrivono le principali API esposte dal Server FHIR.

- GET [base]/Device

Permette di ottenere tutti i device registrati.

- `POST [base]/Device`

Specificando nel body la risorsa FHIR in formato JSON del dispositivo è possibile aggiungere un Device.

- `GET, PUT, DELETE [base]/Device/device_id`

Permette di ottenere un device specifico dato il suo id univoco.

- `GET [base]/DiagnosticReport`

Ottenere tutti i DiagnosticReport registrati.

- `POST [base]/DiagnosticReport`

Aggiungere un nuovo Report come insieme di osservazioni.

- `GET [base]/DiagnosticReport?subject=Device/device_id`

Permette di ottenere tutti i report relativi ad un certo device.

Ad esempio effettuando una chiamata GET su tutti i DiagnosticReport presenti verrà restituita una risorsa di tipo Bundle che altro non è che una risorsa contenitore di altre risorse. Di seguito un esempio di Bundle in formato JSON in risposta alla chiamata descritta.

Listato 4.2: Esempio di risposta alla chiamata API relativa a tutti Device

```

1 {
2   "resourceType": "Bundle",
3   "id": "c52bb290-db74-4331-a134-72a7eb24241f",
4   "type": "searchset",

```

```
5     "total": 1,  
6     "link": [  
7         {  
8             "relation": "self",  
9             "url": "https://localhost:9443/fhir-server/  
10                api/v4/DiagnosticReport?_count=10&_page  
11                =1"  
12         }  
13     ],  
14     "entry": [  
15         {  
16             DiagnosticReport 1,  
17             DiagnosticReport 2,  
18             ...  
19         }  
20     ]  
21 }
```

4.4.3 Approcci differenti

L'intero sistema è stato modellato come un componente che deve essere interrogato per restituire delle informazioni o risorse (polling). Il caso classico infatti vede le applicazioni client richiedere, tramite opportune API, le informazioni di cui hanno bisogno nel momento in cui ne hanno bisogno in modo asincrono. In un'ottica differente si può pensare che all'arrivo di nuovi dati questi vengano notificati direttamente alle applicazioni collegate. Si cerca quindi un sistema reattivo ad eventi dove ogni cambiamento in termini di informazioni presenti viene propagato alle app di interesse. Nello specifico si tratta di un sistema publish-subscribe, dove i subscriber decidono a quali dati sono interessati e si "abbonano" a questi (dichiarano a quali eventi sono interessati), mentre i publisher appena rilevano un cambiamento si occupano di inviarli. Questo approccio permette di avere un sistema asincrono dinamico e scalabile. L'IBM FHIR Server mette per esempio a disposizione un Notification Service che si occupa di inviare una notifica alle app

ogni volta che una risorsa viene creata, aggiornata o eliminata nel FHIR Server. Esso supporta due implementazioni: WebScoekt e Kafka.

Conclusioni e sviluppi futuri

In questa tesi si è affrontato il problema dell'interoperabilità in ambito Internet of Medical Things con caso particolare il dispositivo EGA situato all'ospedale M. Bufalini di Cesena. Si è quindi partiti con una descrizione del sistema esistente, del progetto già in uso Tracking for Care e di quello specifico per il dispositivo T4C EgaService, per arrivare a proporre un componente integrante del servizio in grado di implementare lo standard internazionale per l'interoperabilità tra dispositivi medici HL7 FHIR. Si poteva pensare di implementare la soluzione come un servizio che esponesse delle API a sua volta, in modo da essere completamente intercambiabile, tuttavia si è scelto di implementarlo come componente in quanto fin dal principio visto come parte integrante del progetto T4C già esistente. Il software permette di codificare ogni messaggio HL7 in arrivo dal gateway come risorsa FHIR e di esporre questa risorsa tramite le API del FHIR server. Si tratta quindi di un passo in avanti verso un intero sistema interoperabile, dove ogni dispositivo medico possa scambiare informazioni con gli altri. Rivolti al futuro sarebbe importante capire e decidere a quale livello posizionare il FHIR Server, quindi la gestione delle risorse ad esso collegate. Sarebbe quindi interessante un'analisi dei pro e dei contro, per esempio, di un FHIR Server a livello ospedaliero piuttosto che a livello di reparto o singola stanza. In questo secondo caso ci sarebbe sicuramente un ridondanza di alcune risorse come quella del Patient ma anche una maggiore accuratezza e suddivisione logica del sistema. Nel primo caso invece si avrebbe una visione di insieme di tutte le risorse ma sarebbe anche estremamente complesso gestire ed indicizzare su tutte le risorse presenti. Tra gli sviluppi futuri troviamo l'adattamento delle chiamate API effettuate dalle applicazioni che precedentemente si interfacciavano con il servizio Tracking for Care. Queste infatti

dovranno ottenere le risorse direttamente dal FHIR Server, quindi effettuare delle chiamate Http sulle risorse richieste. Un altro adattamento futuro da valutare sarebbe quello di creare delle API specifiche da esporre che si interfaccino a loro volta con quelle del FHIR Server, in modo da decidere quali risorse esporre e quali no, o quali chiamate permettere di fare.

Ringraziamenti

Voglio ringraziare la mia famiglia per essermi stata accanto e di supporto durante tutto il periodo universitario ed in particolare in quello della tesi. Ringrazio i professori Alessandro Ricci e Angelo Croatti per la disponibilità e il supporto mostrati. Infine ringrazio il collega Elia Urbini con cui ho svolto la parte di tesi integrata.

Bibliografia

- [1] Contributors hapi fhir. <https://hapifhir.io/hapi-fhir/contributors.html>.
- [2] Pagina fhir search. <https://www.hl7.org/fhir/search.html>.
- [3] Point-of-care device implementation guide. <http://build.fhir.org/ig/HL7/uv-pocd/>.
- [4] Iso/iec/ieee international standard - health informatics–device interoperability–part 10201:point-of-care medical device communication–domain information model. *ISO/IEEE 11073-10201:2020(E)*, pages 1–182, 2020.
- [5] Alamsyah, Mery Subito, and Ardi Amir. Design system body temperature and blood pressure monitoring based on internet of things. In *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, pages 276–279, 2020.
- [6] Azana Hafizah Mohd Aman, Wan Haslina Hassan, Shilan Sameen, Zainab Senan Attarbashi, Mojtaba Alizadeh, and Liza Abdul Latiff. Iomt amid covid-19 pandemic: Application, architecture, technology, and security. *Journal of Network and Computer Applications*, page 102886, 2020.
- [7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

- [8] Malti Bansal and Bani Gandhi. Iot amp; big data in smart healthcare (ecg monitoring). In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 390–396, 2019.
- [9] Paolo Bellagente, Alessandro Depari, Paolo Ferrari, Alessandra Flammini, Emiliano Sisinni, and Stefano Rinaldi. M₃/sup₃/sup₃iot — message-oriented middleware for m-health internet of things: Design and validation. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, 2018.
- [10] Yassine Bouteraa and Ismail Ben Abdallah. Development of iot-based robot for wrist rehabilitation. In *2020 17th International Multi-Conference on Systems, Signals Devices (SSD)*, pages 735–739, 2020.
- [11] Ankur Dumka and Anushree Sah. Chapter 9 - smart ambulance traffic management system (satms)—a support for wearable and implantable medical devices. In Nilanjan Dey, Amira S. Ashour, Simon James Fong, and Chintan Bhatt, editors, *Wearable and Implantable Medical Devices*, volume 7 of *Advances in ubiquitous sensing applications for healthcare*, pages 215–228. Academic Press, 2020.
- [12] Ahmed M Elmisery, Seungmin Rho, and Mohamed Aborizka. A new computing environment for collective privacy protection from constrained healthcare devices to iot cloud services. *Cluster Computing*, 22(1):1611–1638, 2019.
- [13] Paul Frehill, Desmond Chambers, and Cosmin Rotariu. Using zigbee to integrate medical devices. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6717–6720, 2007.
- [14] Ali Ghubaish, Tara Salman, Maede Zolanvari, Devrim Unal, Abdulla Al-Ali, and Raj Jain. Recent advances in the internet-of-medical-things (iomt) systems security. *IEEE Internet of Things Journal*, 8(11):8707–8718, 2021.
- [15] Olaronke Iroju, Abimbola Soriyan, Ishaya Gambo, and Janet Olaleke. Interoperability in healthcare: benefits, challenges and resolutions. *International Journal of Innovation and Applied Studies*, 3(1):262–270, 2013.

- [16] S. M. Riazul Islam, Daehan Kwak, MD. Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.
- [17] R. S. H. Istepanian, S. Hu, N. Y. Philip, and A. Sungoor. The potential of internet of m-health things “m-iot” for non-invasive glucose level sensing. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5264–5266, 2011.
- [18] Abdul Jaleel, Tayyeb Mahmood, Muhammad Awais Hassan, Gulshan Bano, and Syed Khaldoon Khurshid. Towards medical data interoperability through collaboration of healthcare devices. *IEEE Access*, 8:132302–132319, 2020.
- [19] Gulraiz Joyia, Rao Liaqat, Aftab Farooq, and Saad Rehman. Internet of medical things (iomt): Applications, benefits and future challenges in healthcare domain. *Journal of Communications*, 12:240–247, 04 2017.
- [20] Martin Kasparick, Stefan Schlichting, Frank Golatowski, and Dirk Timmermann. New ieee 11073 standards for interoperable, networked point-of-care medical devices. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1721–1724, 2015.
- [21] Sarfraz Fayaz Khan. Health care monitoring system in internet of things (iot) by using rfid. In *2017 6th International Conference on Industrial Technology and Management (ICITM)*, pages 198–204, 2017.
- [22] Mengyang Li, Heather Leslie, Bin Qi, Shan Nan, Hongshuo Feng, Hailing Cai, Xudong Lu, Huilong Duan, et al. Development of an openehr template for covid-19 based on clinical guidelines. *Journal of medical Internet research*, 22(6):e20239, 2020.
- [23] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, 4(5):1125–1142, 2017.

- [24] D Liveri. Smart hospitals-security and resilience for smart health service and infrastructures. *European Union Agency For Network And Information Security*, 2016.
- [25] Ing. Antonio De Mori. La storia dell'emogas. <https://www.area-c54.it/public/storia%20dell'emogasanalisi.pdf>.
- [26] Amit Palve and Harshita Patel. Towards securing real time data in iomt environment. In *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 113–119, 2018.
- [27] Warish D. Patel, Sharnil Pandya, Baki Koyuncu, Bhupendra Ramani, Sourabh Bhaskar, and Hemant Ghayvat. Nxtgeuh: Lorawan based next generation ubiquitous healthcare system for vital signs monitoring amp; falls detection. In *2018 IEEE Punecon*, pages 1–8, 2018.
- [28] Pallavi Patil and Kalyani Bhole. Real time ecg on internet using raspberry pi. In *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pages 267–270, 2018.
- [29] Karen Rose, Scott Eldridge, and Lyman Chapin. The internet of things: An overview. *The internet society (ISOC)*, 80:1–50, 2015.
- [30] B. Sargunam and S. Anusha. Iot based mobile medical application for smart insulin regulation. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–5, 2019.
- [31] Pallavi Sethi and Smruti R Sarangi. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 2017.
- [32] S. Sivaraj, K. Vigneshwaran, S. Vigneshwaran, and M. Priyan. Iot ambulance with automatic traffic light control. 2017.
- [33] Mir Sajjad Hussain Talpur. The appliance pervasive of internet of things in healthcare systems, 2013.

- [34] Treccani. Telemedicina.
- [35] Dong Zhong, Zhu Yian, Wang Lanqing, Duan Junhua, and He Jiaxuan. Continuous blood pressure measurement platform: A wearable system based on multidimensional perception data. *IEEE Access*, 8:10147–10158, 2020.