

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

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

**SVILUPPO DI SISTEMI INTEROPERABILI
NELL'AMBITO DI INTERNET OF MEDICAL
THINGS BASATI SU STANDARD
FHIR: UN CASO DI STUDIO**

Tesi di laurea in
SISTEMI EMBEDDED ED INTERNET OF THINGS

Relatore

Prof. ALESSANDRO RICCI

Candidato

ELIA URBINI

Correlatore

Dott. Ing. ANGELO CROATTI

Anno Accademico 2020-2021

*”Il primo passo è stabilire che qualcosa è possibile; allora si verificherà la
probabilità.”
(Elon Musk)*

Indice

Introduzione	xiii
1 Quadro generale dell'IoMT	1
1.1 Dall'IoT all'IoMT	1
1.2 Architettura IoT	2
1.2.1 Objects layer	3
1.2.2 Objects abstraction layer	4
1.2.3 Service management layer	5
1.2.4 Application layer	6
1.2.5 Business layer	7
1.3 Protocolli e tecnologie abilitanti	7
1.4 Dispositivi dell'IoMT	12
1.4.1 Dispositivi indossabili	13
1.4.2 Dispositivi impiantabili	14
1.4.3 Dispositivi ambientali	14
1.4.4 Dispositivi stazionari	15
1.5 Applicazioni dell'IoMT	15
1.6 Sfide e problemi nell'IoMT	18
1.6.1 Gestione dei dati	18
1.6.2 Scalabilità	19
1.6.3 Design interfaccia utente	19
1.6.4 Sicurezza e privacy	20
1.6.5 Interoperabilità e standardizzazione	20

2	Interoperabilità ecosistema IoMT	21
2.1	Integrating the Healthcare Enterprise	22
2.2	ISO/IEEE 11073	22
2.2.1	ISO/IEEE 11073 PHD	23
2.2.2	ISO/IEEE 11073 SDC	23
2.3	Health Level Seven International	24
2.3.1	HL7 v2	24
2.3.2	HL7 v3	25
2.4	HL7 FHIR	26
2.4.1	Sviluppo di FHIR	26
2.4.2	Cos'è FHIR	27
2.4.3	FHIR come API	28
2.4.4	Estensioni e profili	29
2.4.5	Standard terminologici: LOINC e SNOMED	30
2.5	Altri standard	31
2.5.1	Continua Health Alliance	31
2.5.2	oneM2M	31
2.5.3	ZigBee Health Care	31
2.6	Point-of-Care Device Implementation Guide	32
2.6.1	Panoramica modello generale	33
3	Caso di studio: monitor parametri vitali	35
3.1	Stato dell'arte	35
3.1.1	Cos'è un monitor parametri vitali	35
3.1.2	Infinity M540	37
3.1.3	Analisi dispositivi in commercio	38
3.2	Modellazione con FHIR	39
3.2.1	Device	39
3.2.2	DeviceMetric	40
3.2.3	Observation	40
3.3	Secondo l'Implementation Guide	41
3.3.1	Medical Device System	42
3.3.2	Virtual Medical Device	42

3.3.3	Channel Device	43
3.3.4	Device Metric	45
3.3.5	Observation	50
3.3.6	Conclusioni sul modello ottenuto	50
4	Proposta di integrazione T4C	53
4.1	Stato attuale: T4C e monitor parametri vitali	53
4.2	Proposta di integrazione verso un sistema FHIR interoperabile . . .	56
4.2.1	FHIR Server	56
4.2.2	Specifica dei requisiti	57
4.2.3	Scenario sistema integrato	57
4.3	Progettazione	59
4.4	Sviluppo e testing	61
4.4.1	Tecnologie utilizzate	61
4.4.2	Testing	63
4.4.3	Ottica ad eventi	66

Elenco delle figure

1.1	Architetture IoT proposte [23]	3
1.2	Sistema di monitoraggio pazienti [6]	13
1.3	Classificazione applicazioni IoMT	16
2.1	Diagramma di sequenza FHIR API	28
2.2	Modello PoCD [3]	33
3.1	Esempio di applicazione monitor a parametri vitali [17]	36
3.2	A monitor M540, B stazione di aggancio M500	37
3.3	Modello semplificato con risorse FHIR	41
3.4	ISO/IEEE 11073 SDC modellato in HL7 FHIR [2]	42
3.5	Diagramma UML delle risorse VMD	43
3.6	Diagramma UML canali pressione sanguigna	44
3.7	Diagramma UML canali temperatura corporea	44
3.8	Diagramma UML canali pulsossimetro	45
3.9	Diagramma UML canali ECG	45
3.10	Diagramma UML metriche Invasive Blood Pressure	46
3.11	Diagramma UML metriche Non-Invasive Blood Pressure	47
3.12	Diagramma UML metriche Temperature	47
3.13	Diagramma UML metriche Pulse-Oximeter	48
3.14	Diagramma UML metriche Respiratory Rate	48
3.15	Diagramma UML metriche Heart Rate	49
3.16	Diagramma UML metriche Arrhythmia	49
3.17	Diagramma UML metriche ECG	50
3.18	Diagramma UML complessivo monitor parametri vitali	51

4.1	Diagramma di sequenza scambio dati con il servizio T4C	55
4.2	Diagramma di sequenza sistema integrato	58
4.3	Schema UML componente integrato a T4C	59
4.4	Schema UML modellazione risorse FHIR	60
4.5	Diagramma UML finale del sistema	61

Elenco dei listati

2.1	Esempio messaggio HL7 v2	25
2.2	Esempio risorsa base FHIR	27
4.1	Esempio osservazione in formato JSON caricata sul server	63
4.2	Esempio risposta del server alla chiamata API per ottenere tutti i monitor	65

Introduzione

L'Internet of Things (IoT) sta cambiando le nostre vite in un modo mai immaginato prima. A differenza del paradigma tradizionale, nel mondo dell'IoT tutte le cose sono considerate come oggetti intelligenti, i quali sono connessi l'uno con l'altro. L'IoT è definito come una rete dinamica di cose fisiche e virtuali dotata di protocolli di comunicazione e standard. Questi oggetti hanno identità e attributi, sono in grado di connettersi a reti come Internet, e possono eseguire rilevamento ed elaborazione di dati, formare reti e comunicare tra di loro. Pertanto, l'IoT può essere considerato come una nuova versione di tecnologia dell'informazione e della comunicazione (TIC).

Spaziando dagli elettrodomestici intelligenti alle città intelligenti, l'IoT ha anche aperto una nuova sfida nel settore sanitario, chiamata Internet of Medical Things (IoMT), anche detta Internet of Healthcare Things (IoHT). Essa gioca un ruolo fondamentale nell'aggiornamento degli ospedali verso il modello di smart hospital, ovvero ospedale intelligente. L'IoMT offre vantaggi significativi per il benessere delle persone, migliorando la qualità della vita e riducendo le spese mediche. Gli elementi critici di questo nuovo paradigma sono i sensori wireless, che possono essere utilizzati per monitorare a distanza lo stato della salute dei pazienti e le tecnologie di comunicazione sfruttate per inviare le informazioni agli operatori sanitari.

Il passo principale verso un ecosistema sanitario intelligente consiste nell'utilizzare il potenziale delle tecnologie già esistenti per fornire migliori servizi agli utenti e migliorare la loro vita. L'intelligenza artificiale è una tecnologia che può essere di supporto all'IoMT, assistendo i medici nel processo decisionale clinico. Inoltre, con le tecniche di Machine Learning e Deep Learning, i computer possono

imparare a prendere decisioni utilizzando i dati generati dagli operatori sanitari e dai feedback dei pazienti stessi.

I dispositivi dell'IoMT supportati dall'Intelligenza Artificiale possono monitorare continuamente la salute delle persone. Robot intelligenti, case intelligenti e assistenti virtuali possono fornire il supporto necessario a persone anziane e disabili. Combinando le informazioni raccolte dai sensori e le informazioni ottenute dal sistema sanitario, è possibile monitorare e prevenire determinate malattie. Durante i disastri, i sistemi intelligenti possono aiutare le autorità a fornire la giusta assistenza alle persone e ad adottare le misure necessarie in modo tempestivo.

Nel contesto dell'IoMT, i componenti di input come biosensori, moduli di comunicazione e il personale lavorano in modo cooperativo per fornire il miglior servizio sanitario in modo efficiente e sicuro. Con l'ausilio delle tecnologie IoMT, la cura autonoma e la diagnosi precoce sono considerati i servizi che contribuiscono maggiormente al rafforzamento dell'ecosistema sanitario, in particolare quelli che utilizzano sistemi di monitoraggio remoto. Quest'ultimo dipende in modo cruciale dall'elaborazione e dall'analisi delle informazioni in tempo reale raccolte tramite biosensori. Lo scambio di dati tra tali dispositivi richiede meccanismi sicuri e tecnologie di comunicazione. D'altra parte, la perdita di dati e il furto di informazioni sono problemi seri, se questi dispositivi non sono adeguatamente protetti.

I dispositivi usati nell'IoMT sono generalmente interoperabili e possono essere connessi a un unico sistema scalabile quando appartengono allo stesso venditore. Questa dipendenza, tuttavia, presenta un collo di bottiglia in quasi tutti gli scenari di utilizzo pratico a causa di una vasta gamma di sensori e strumenti medici utilizzati in ambito ospedaliero al giorno d'oggi. Data la sua massima importanza, l'interoperabilità rimane al centro di numerose ricerche recenti.

Quindi, partendo da questa criticità, l'obiettivo di questa tesi è di fare una proposta di integrazione efficace per il progetto Tracking for Care (T4C), in collaborazione con AUSL della Romagna e Ospedale Bufalini, e in particolare del dispositivo monitor a parametri vitali.

Nel primo capitolo si cercherà di delineare un quadro generale dell'IoT e della sua applicazione nel settore sanitario con l'IoMT. Proseguendo, nel secondo capitolo si porrà l'attenzione sul problema dell'interoperabilità introducendo al lettore

i vari standard oggi presenti tra cui FHIR, quello adoperato nei capitoli successivi. Infatti, nel terzo capitolo, a partire da una specifica di FHIR, si procederà a modellare il dispositivo in esame, il monitor parametri vitali, sfruttando le risorse messe a disposizione da FHIR, dopo aver fatto una breve panoramica sul modello in uso presso l'ospedale Bufalini e agli altri in commercio. Infine, nel quarto capitolo verrà presentata una possibile soluzione al problema dell'interoperabilità attraverso un modulo che, integrato al progetto T4C, permetta di rappresentare e gestire le informazioni rilevate dal monitor secondo un formato conforme allo standard FHIR.

Capitolo 1

Quadro generale dell'IoMT

L'enorme progresso nei domini dell'elettronica e dei sistemi di comunicazione, in particolare di quella wireless, nello scorso decennio, ha rafforzato la mobilità dei dispositivi e i servizi onnipresenti per gli utenti. Questo ha dato impulso all'idea di collegare il mondo fisico a quello virtuale promuovendo l'emergere di "Cyber-Physical-Systems" (CPS) e, di conseguenza, "Internet of Things" (IoT).

In questo capitolo verrà introdotto il contesto generico nel quale si cala questa tesi di laurea a partire da una breve definizione dell'Internet delle Cose. Quindi, si procederà esaminando lo stato dell'arte dell'Internet of Medical Things (IoMT) a cominciare dai modelli architetturali e dai protocolli in uso, continuando con una panoramica dei vari dispositivi intelligenti impiegati nel settore medico. Proseguendo, verranno descritti gli scenari applicativi più comuni dell'IoMT concludendo con le principali sfide e problemi che devono essere affrontati al giorno d'oggi tra cui l'interoperabilità, argomento centrale del caso di studio.

1.1 Dall'IoT all'IoMT

L'Internet of Things (IoT) copre molte aree, tuttavia la sua definizione rimane ancora vaga e troppo ampia. Nel tentativo di dare una solida definizione per IoT, IEEE Internet Initiative ha pubblicato un report intitolato "Towards a definition of the Internet of Things (IoT)" [21] dove viene descritto come un dominio applicativo che integra diversi ambiti tecnologici e sociali.

L'IoT è una nuova tendenza nel campo della computazione con una filosofia alternativa rispetto a quella tradizionale, consentendo l'interazione tra il mondo fisico e quello cibernetico. L'idea alla base del successo del paradigma IoT è quella di sviluppare, integrare e interconnettere moduli intelligenti per realizzare sottosistemi dai compiti isolati, quindi estenderli per fare in modo che un sistema globale più grande persegua molteplici obiettivi.

Sono le applicazioni, la facilità di implementazione delle stesse e i domini che può coprire, a rendere una determinata tecnologia interessante e persistente. L'IoT si estende a un'ampia gamma di domini applicativi quali: produzione, monitoraggio di infrastrutture, monitoraggio dell'ambiente, sistemi di alimentazione, assistenza sanitaria e usi personali o letteralmente qualsiasi cosa si possa pensare. In pratica tutto può essere una "Thing". Un approccio tipico è quello di incorporare sensori negli oggetti di uso quotidiano rendendoli accessibili online, consentendo un monitoraggio preciso e un processo decisionale guidato dagli eventi [23].

Il crescente sviluppo di sensori in ambito medico e sanitario ha creato un'opportunità di sfruttare l'IoT per la raccolta e l'analisi dei dati medici. L'utilizzo combinato di dispositivi intelligenti e di sensori diagnostici portatili ha aperto la strada a un ramo dell'IoT chiamato Internet of Medical Things (IoMT) o Internet of Healthcare Things (IoHT). L'IoMT può essere definito come la connessione di dispositivi medici a un sistema sanitario attraverso una rete online, come un cloud, che spesso prevede comunicazione macchina-macchina.

L'uso dell'IoT per applicazioni mediche è tuttavia ancora in fase di sviluppo. Attualmente, l'IoMT è spesso utilizzato per le attività di monitoraggio a distanza dei pazienti con malattie croniche così come in ambito ospedaliero.

1.2 Architettura IoT

Poiché l'IoT è un dominio molto ampio, sia a livello locale che globale, le aspettative per le parti interessate (sia gli utenti finali che le industrie coinvolte nello sviluppo e nella produzione) sono elevati in termini di servizi e opportunità. L'elenco dei domini applicativi è enorme e siccome l'IoT è una fusione di più tecnologie, che consente l'interazione tra questi, richiede un'architettura flessibile a strati. In

vista di questo problema, diversi gruppi di ricerca, il mondo accademico, industriale e organizzazioni di standardizzazione come "IEEE", "ETSI", "IETF", "W3C" e molte altre hanno proposto le loro architetture in base alle loro visioni e percezioni. Nonostante ciò, non è ancora stato delineato un modello di riferimento comune.

Nella seguente sezione, verranno descritti alcuni dei modelli proposti in letteratura e verrà fornita una breve discussione sul modello a cinque strati.

Il modello architetturale base per l'IoT è suggerito da Project IoTA, sulla base dello studio sui requisiti comuni posti dalla ricerca e dalle organizzazioni industriali. L'architettura è a tre strati e comprende: application layer (livello applicativo), network layer (livello di rete) e perception layer (livello fisico o di percezione). IEEE propone modelli simili ma attribuendo nomi leggermente diversi ai tre strati. Alcune pubblicazioni recenti, tuttavia, hanno proposto modelli differenti che astraggono maggiormente l'architettura IoT con altri livelli. La Figura 1.1 mostra alcuni dei modelli architetturali IoT proposti, tra questi quello a cinque strati è il più generico e può risultare adatto per molteplici scenari IoT [23].

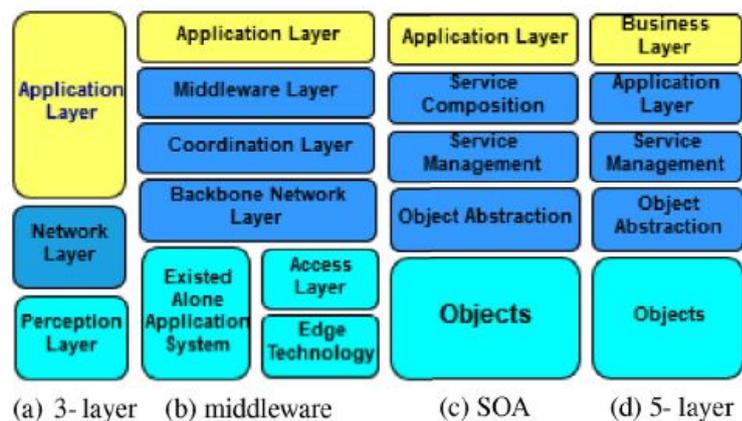


Figura 1.1: Architetture IoT proposte [23]

1.2.1 Objects layer

L'"objects layer", noto anche come livello di percezione è il primo livello nel modello architetturale IoT a cinque strati. Questo livello è costituito da sensori,

attuatori e dispositivi di generazione di dati come RFID, NFC o infrarossi, che misurano o rilevano l'ambiente circostante. I dispositivi sono generalmente collegati tramite reti denominate "Wireless Sensor Networks" (WSN) o "Wireless Sensor and Actuation Networks" (WSAN).

Questo livello è simile a quello fisico del modello TCP/IP ed è responsabile della generazione dei dati. Tuttavia, non bisogna confondersi, poiché il modello TCP/IP è diverso e viene utilizzato come architettura Internet e non IoT. In aggiunta, questo livello fornisce servizi come l'identificazione dei dispositivi, e una raccolta di informazioni specifiche relative ai sensori che possono essere richieste a fini gestionali. Le informazioni raccolte vengono solitamente trasmesse attraverso la rete verso sistemi centrali di elaborazione (cloud o locali al dominio).

Poiché lo strato di percezione è costituito da dispositivi/oggetti eterogenei, questi devono essere configurati per essere in grado di svolgere i loro compiti. Inoltre, questo livello deve essere scalabile, ad esempio sfruttando meccanismi plug-and-play standardizzati. Altre responsabilità del livello di percezione includono la digitalizzazione e il trasferimento dei dati all'"Objects abstraction layer" attraverso canali sicuri. L'enorme quantità di dati, cioè i big data hanno inizio a questo livello [23].

1.2.2 Objects abstraction layer

In informatica, un livello di astrazione consiste in un modo di nascondere i dettagli implementativi di un sottosistema, consentendo la separazione di compiti per facilitare l'interoperabilità e l'indipendenza dalla piattaforma. Siccome nell'IoT sono coinvolti molti dispositivi e tecnologie eterogenee, c'è un enorme aumento di complessità nella gestione e identificazione a livello di dispositivo o piattaforma. Il compito dell'"object abstraction layer" consiste nel ridurre la complessità fornendo astrazione su diversi livelli. Ad esempio, l'astrazione a livello di struttura del software, consente ai programmatori di applicazioni di sviluppare moduli software senza considerare le complessità architettoniche degli oggetti. È un po' più complicato da implementare in ambiente IoT, dal momento che i programmatori hanno a che fare con i sottosistemi IoT distribuiti attraverso reti di dispositivi eterogenei. In un simile scenario "Open Distributed Object Framework" (OpenDOF), che è

un adattamento del paradigma object-oriented per i sistemi distribuiti, può essere un ottimo supporto. Il framework gestisce l'overhead per i meccanismi di comunicazione e sicurezza, consentendo ai programmatori di concentrarsi sullo sviluppo di soluzioni IoT astruendo dai dispositivi stessi.

L'identificazione degli oggetti è cruciale per la fruizione di servizi nell'IoT. Con milioni di dispositivi, l'identificazione e l'autenticazione risultano essere significative per la sicurezza, prevenendo attacchi di vario genere. Soddisfare questi requisiti per l'ecosistema IoT è una grande sfida e compito dell'"object abstraction layer". In questo contesto, "GlobalSign", un'azienda tecnologica che fornisce soluzioni affidabili di identificazione e sicurezza, ha trovato una soluzione offrendo l'identificazione del dispositivo e l'autenticazione come servizio basato su cloud.

I sistemi IoT sono ad alta intensità di comunicazione, ovvero una moltitudine di messaggi devono essere comunicati attraverso una serie di tecnologie e protocolli. Tuttavia, mancano standard reali con cui lavorare, il che rende l'ambiente di sviluppo molto più impegnativo. L'astrazione a livello di comunicazione aiuta i programmatori fornendo un singolo livello di astrazione per diverse tecnologie di comunicazione (ad esempio RFID, BLE, infrarossi, ZigBee ecc.) e gestendo vari requisiti di sicurezza.

Altre funzioni di questo livello includono l'astrazione di piattaforme del livello applicativo. I servizi nell'IoT sono solitamente basati su cloud di diversi fornitori. Queste piattaforme possono essere eterogenee e fornire API diverse di supporto per lo sviluppo di applicazioni differendo spesso in politiche di sicurezza e gestione dei dati. Tenere traccia di tutto ciò risulta essere un grosso sovraccarico durante lo sviluppo delle applicazioni. L'"object abstraction layer" fornisce una piattaforma end-to-end garantendo astrazione per varie piattaforme cloud e supportando la gestione dei dati e altri requisiti in modo trasparente [23].

1.2.3 Service management layer

Le applicazioni e i dispositivi IoT richiedono diversi tipi di servizi resi disponibili da specifici fornitori. Quindi, meccanismi di scoperta e mappatura dei servizi sono necessari in modo che i "subscribers" (o abbonati) possano trovare un determinato servizio per poi sottoscrivere al fornitore. Inoltre, potrebbero esserci problemi

di interoperabilità tra l'abbonato e il provider, rendendo difficile l'interazione. Tutte queste funzioni sono gestite dal "service management layer" chiamato anche middleware. Il livello fornisce un sistema di mappatura per l'associazione tra i richiedenti di servizi e i fornitori di servizi. Questo strato fornisce anche un livello di astrazione o middleware per oggetti eterogenei in modo che gli sviluppatori di applicazioni non devono considerare le specifiche hardware. In breve, le funzioni supportate dal service management layer sono [23]:

- ricerca e gestione di servizi IoT fornendo funzionalità di mappatura;
- protezione di dati, dispositivi e analisi delle informazioni;
- fornire una piattaforma middleware agli sviluppatori di applicazioni eseguendo astrazione rispetto all'hardware;
- gestione dei dati per estrarre informazioni da enormi quantità di dati grezzi generati dal livello di percezione;
- gestione del flusso di dati e fornitura di un'interfaccia tra reti cablate e reti wireless;
- funzioni di elaborazione dati come data mining, text mining ecc.

1.2.4 Application layer

Nel modello di architettura a cinque livelli proposto, il livello applicativo lavora sopra al livello di rilevamento e gestione dei servizi ed è responsabile della fornitura di servizi intelligenti agli utenti finali come un essere umano, un sistema fisico o un processo. Secondo la gerarchia, il livello applicativo è il secondo più alto nel modello suggerito, tuttavia dal punto di vista dell'utente finale è quello superiore, fornendo un'interfaccia per l'interazione tra i servizi e l'utente. Un altro ruolo importante di questo livello è l'utilizzo efficace dei dati raccolti dai livelli inferiori presentandoli all'utente finale in una forma interpretabile [23].

1.2.5 Business layer

Il business layer lavora sopra tutti gli altri livelli ed è responsabile della gestione di tutte le attività del sistema, di costruire modelli di business, report, diagrammi di flusso ecc. Dovrebbe anche analizzare dati a supporto del processo decisionale, confrontando l'output di ogni livello con l'output previsto per migliorare i servizi. Fondamentalmente tutte le funzioni di alto livello, analisi dei dati, valutazione dell'output e sicurezza così come il monitoraggio dei sottosistemi sono realizzate in questo livello [23].

1.3 Protocolli e tecnologie abilitanti

In questa sezione verranno descritti i principali protocolli e tecnologie abilitanti nell'IoT con occhio critico per quanto riguarda il loro utilizzo nel dominio sanitario e quindi nell'IoMT. Di seguito una lista a partire dalle tecnologie più vicine al livello hardware fino a quelle del livello applicativo.

- **Infrarossi:** la tecnologia a infrarossi proposta dalla Infrared Data Association (IrDA) utilizza la luce a infrarossi per la comunicazione a corto raggio. IrDA fornisce linee guida per protocolli di comunicazione che si basano sulla tecnologia wireless a infrarossi. Lo stack del protocollo IrDA è composto da tre livelli obbligatori: fisico, link access (IrLAP) e link management (IrLMP). Il protocollo IrLAP fornisce il livello di collegamento di base tra coppie di dispositivi e si basa sullo standard High-level Data Link Control (HDLC) per la creazione della connessione e il trasferimento dei dati. Il protocollo IrLMP fornisce un modo per più entità all'interno di coppie di dispositivi IrDA di utilizzare contemporaneamente e indipendentemente la singola connessione IrLAP.

All'interno del dominio medico, IrDA viene utilizzato in sensori di temperatura, in termometri e fotocamere. La tecnologia di imaging termico a infrarossi viene utilizzata anche per fotografare un corpo e trovarne la temperatura locale [22].

- RFID: l'identificazione a radiofrequenza (RFID) è una tecnologia di identificazione di oggetti wireless che utilizza segnali a radiofrequenza per comunicazioni a corto raggio. Le entità principali nella comunicazione RFID sono due: un dispositivo di lettura (il lettore RFID) e l'RF tag. Il lettore memorizza i dati e dispone di funzioni di lettura a distanza. Il tag inizialmente riceve un messaggio dal lettore, e risponde inviando un identificativo e/o altre informazioni al lettore stesso.

Esistono due tecnologie principali per i tag RFID: tag a lettura attiva e tag a lettura passiva. Quelli a lettura attiva sono alimentati, utilizzano bande ad alta frequenza e sono relativamente costosi; d'altra parte, i tag passivi di solito operano su frequenze più basse e non hanno una fonte di alimentazione interna.

I tag RFID autonomi posizionati internamente o in prossimità al corpo di un paziente, sono tra le tecnologie che permettono lo sviluppo di sistemi sanitari con monitoraggio dell'ambiente in cui si trovano i pazienti. L'RFID è stato utilizzato per rilevare gli accessi all'interno dei vari luoghi. Un altro esempio di uso dell'RFID è presentato in [14], dove gli autori utilizzano tag e sensori per fornire un monitoraggio preciso della temperatura per lo stoccaggio dei farmaci regolando la temperatura in base al tipo di farmaco.

- NFC: Il protocollo Near Field Communication (NFC) supporta la comunicazione a corto raggio tra dispositivi (nel raggio di pochi centimetri al massimo). Come nel caso dell'RFID, anche l'NFC opera sia in modalità passiva che attiva. NFC viene utilizzato per una connessione facile e a basso costo tra dispositivi IoT. Il più delle volte, questo protocollo viene utilizzato per scopi di autenticazione; per esempio, deve permettere di controllare se un dispositivo IoT è registrato in un server di autenticazione, nonché verificare se l'utente è registrato nel server stesso.

In [16] viene esaminato l'uso dei segnali NFC per i dispositivi medici all'interno del corpo di un essere umano (ad es. sensori ingeribili o impiantabili). L'uso di NFC all'interno di dispositivi medici impiantati o ingeriti presenta tuttavia diverse sfide di biocompatibilità.

- **Bluetooth/BLE:** Bluetooth è una tecnologia di comunicazione wireless diffusa, basata sullo standard IEEE 802.15.1. È adatta per dispositivi a bassa potenza e a basso costo e funziona sulla banda a 2,4 GHz. È idonea per la trasmissione di dati tra dispositivi mobili a corto raggio, che può variare in base alla versione. La sua versione a basso consumo è nota come Bluetooth Low Energy (BLE) o Bluetooth Smart. BLE fornisce velocità di trasmissione dati più elevata rispetto ad altri protocolli wireless a bassa potenza, come ZigBee o Z-Wave discussi in seguito.

Viene utilizzato per applicazioni IoMT che richiedono comunicazioni a breve distanza, applicazioni sportive, dispositivi fitness e dispositivi medici portatili.

- **Z-Wave:** Z-wave è un protocollo MAC wireless a bassa potenza sviluppato da Zensys. Può essere utilizzato per impostare reti con topologia a maglia e supporta due tipi di dispositivi, dispositivi di controllo e dispositivi slave. Z-wave è adatto a supportare brevi messaggi tra i dispositivi IoT utilizzati per luce, energia e controllo sanitario. Viene utilizzato nell'automazione domestica wireless, soprattutto in smart home, aree sanitarie e piccoli domini commerciali.
- **UWB:** costruito sullo standard IEEE 802.15.3, l'obiettivo principale del protocollo Ultra-Wideband (UWB) è quello di supportare comunicazioni wireless in ambienti chiusi ad alta velocità e a corto raggio. Data la sua elevata velocità che può variare da 110 a 480 Mbps, trova spazio in applicazioni audio e video all'interno della rete domestica. Grazie al suo basso consumo e alta precisione, UWB è adatto per applicazioni in tempo reale in determinati luoghi come gli ospedali. Un uso tipico è quello di trasmettere segnali da sensori a un microcontrollore. Per esempio, la procedura dell'elettrocardiogramma richiede una tecnologia di comunicazione a breve distanza e UWB, tra gli altri protocolli, è stato utilizzato per questo scopo [7].
- **WiFi:** Wireless Fidelity (WiFi) è un protocollo a medio raggio (fino a 100 m), basato sulla famiglia di standard dell'IEEE 802.11. Il WiFi è ampiamente utilizzato per dispositivi palmari e per reti locali, supportando l'accesso

a Internet per più dispositivi. Attualmente, lo standard WiFi più comunemente utilizzato è 802.11n, che può supportare una velocità di trasmissione di centinaia di Mbps, rendendolo ottimale per il trasferimento di file. Tuttavia, non è sempre adatto per applicazioni IoT, a causa del suo consumo energetico relativamente elevato.

Diversi autori hanno proposto l'utilizzo del WiFi per la comunicazione dei dispositivi di monitoraggio in un sistema IoMT. Ad esempio [11] usa il WiFi su una rete di 45 dispositivi medici, comprese pompe per infusione, defibrillatori, ventilatori polmonari e macchine per anestesia, a dimostrare che il WiFi può essere applicato in modo efficace e sicuro nella comunicazione di questi dispositivi.

Per la compatibilità IoT, l'alleanza WiFi ha proposto WiFi HaLow, che opera nello spettro al di sotto di un GHz ed è basato sullo standard IEEE 802.11ah. In [15] viene proposto proprio questo protocollo per la trasmissione dati all'interno di una Wireless Body Area Network (WBAN).

- ZigBee: ZigBee è un protocollo di comunicazione conforme a IEEE 802.15.4, a basso costo, bassa velocità e basso consumo. ZigBee può essere utilizzato in vari ambienti IoT. È ampiamente utilizzato nelle aree sanitarie tanto che, nel 2009, la ZigBee Alliance ha introdotto il profilo di applicazione pubblica ZigBee Health Care. Il profilo ZigBee Health Care si basa su ZigBee Pro e implementa un protocollo a livello applicativo completamente funzionante per gli ambienti sanitari.
- WIA-PA: secondo [29], WIA-PA è uno standard di comunicazione wireless industriale cinese. È stato proposto per sostituire standard come IEEE 802.15.1 e IEEE 802.15.4. WIA-PA è progettato per la misurazione, il monitoraggio e il controllo di processi di produzione. WIA-PA rispetto agli altri standard (wireless HArt, ISA SP100) è in grado di soddisfare il real-time e l'affidabilità in sistemi su base industriale. Tuttavia, in [24] propongono questo protocollo "industriale" per uso medico in un sistema di monitoraggio remoto come protocollo di trasmissione nella rete di sensori wireless.

- ISA 100.11a: l'obiettivo principale del comitato ISA100 è quello di portare avanti una serie di standard che coprano reti wireless industriali. In particolare, questi standard mirano a soddisfare le esigenze dell'intero ciclo di produzione. ISA-100.11a, il primo standard della famiglia, propone una rete con topologia a maglia che mira a fornire comunicazioni wireless sicure per il controllo dei processi. In [24] propongono l'uso di questo protocollo similmente a quanto fatto per WIA-PA.
- 6LoWPAN: sviluppato dall'Internet Engineering Task Force (IETF), LoWPAN è un protocollo wireless con larghezza di banda ridotta, dimensioni dei pacchetti limitate e lunghezza dell'indirizzo variabile, utilizzato principalmente per consentire ai dispositivi IoT di unirsi alle reti IP. 6LoWPAN è la sua versione IPv6. Nel caso del settore sanitario, i sensori IoMT e i dispositivi locali possono essere collegati alle reti IP tramite 6LoWPAN, consentendo l'interconnessione tra un gruppo di sensori. Inoltre, consente l'interconnessione di sensori con dispositivi middleware o router connessi a Internet.
- LoRaWAN: LoRa (Long Range) è un protocollo di livello fisico originariamente sviluppato per supportare reti a bassa potenza e WAN. Siccome LoRa è definito a livello fisico, è necessario che i protocolli del livello di rete superiore siano costruiti sopra di esso. LoRaWAN è un protocollo di livello MAC, che agisce principalmente come protocollo di livello di rete. Essenzialmente, gestisce il routing e la comunicazione tra gateway e terminali. LoRaWAN si concentra sulle applicazioni WAN (Wide Area Network) ed è progettato per consentire un basso consumo con funzionalità specificamente necessarie per supportare comunicazioni a basso costo, bidirezionali e sicure in varie applicazioni come IoT, Machine-to-Machine (M2M), smart city e applicazioni industriali. Ad esempio, in [20] viene presentato un sistema di monitoraggio sanitario remoto in cui viene utilizzata un'infrastruttura LoRaWAN per trasmettere i dati raccolti.
- COAP: il Constrained Application Protocol (COAP) è un protocollo web di trasferimento, particolarmente adatto ai nodi IoT con memoria e potenza di

elaborazione limitate per questo funziona molto bene in ambienti sanitari. È costruito sul protocollo UDP e fornisce un meccanismo leggero per supportare una comunicazione affidabile. Utilizza un'architettura Representational State Transfer (REST) molto più leggera del tipico protocollo HTTP. [27] descrive un semplice scenario IoMT proponendo COAP come protocollo a livello applicativo per un sistema di monitoraggio remoto.

- MQTT: Message Queue Telemetry Transport (MQTT) è stato sviluppato da IBM, con l'obiettivo principale di supportare comunicazioni Machine to Machine (M2M) leggere. È un protocollo a scambio di messaggi asincrono con un'architettura publish/subscribe. Il suo obiettivo è di fornire larghezza di banda ed efficienza dal punto di vista del consumo energetico, rendendolo così adatto a dispositivi con memoria e capacità computazionale limitate. In [27] viene descritto uno scenario IoMT e proposto MQTT come protocollo a livello applicativo per la comunicazione tra il cloud e i dispositivi dell'utente finale come un telefono cellulare.
- HTTP: In diversi casi, il semplice HTTP viene utilizzato a livello applicativo. [27] descrive uno scenario IoMT che utilizza HTTP per la comunicazione tra il cloud e gli operatori sanitari. Anche [14] usa HTTP in un sistema sanitario provvisto di un termometro indossabile e un sensore a infrarossi.

1.4 Dispositivi dell'IoMT

Il primo passo verso un sistema sanitario intelligente prevede l'acquisizione e la collezione di dati (e.g., temperatura corporea, frequenza del battito cardiaco) usando attrezzatura specifica (i.e., sensori), i quali andranno poi trasferiti a livello di rete. Ad esempio, nei sistemi di monitoraggio della salute, molti sensori comunicano per garantire che il paziente sia sempre monitorato e possa ricevere aiuto quando necessario il prima possibile come illustrato nella Figura 1.2. Al giorno d'oggi non esiste una classificazione uniforme per quanto riguarda le "medical things", tuttavia esse possono essere suddivise in quattro categorie: dispositivi indossa-

bili (wearable devices), dispositivi impiantabili (implantable devices), dispositivi ambientali (ambient devices) e dispositivi stazionari (stationary devices) [6].

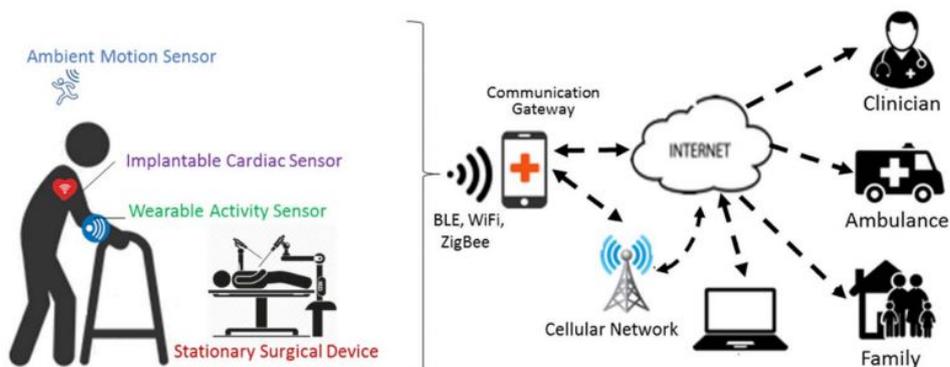


Figura 1.2: Sistema di monitoraggio pazienti [6]

1.4.1 Dispositivi indossabili

I dispositivi indossabili possono essere suddivisi in tre tipologie con caratteristiche e funzioni specifiche in [6]:

- **Gestione della malattia:** questi dispositivi sono progettati per monitorare i pazienti con malattie croniche come asma, malattie cardiache o diabete e hanno lo scopo di migliorare lo stato di salute o assistere nel trattamento di pazienti a cui è già stata diagnosticata una particolare malattia.
- **Previsione della malattia:** questi dispositivi vengono utilizzati per monitorare le informazioni fisiologiche che possono essere utilizzate per impedire agli individui di sviluppare una malattia non ancora diagnosticata.
- **Monitoraggio delle attività:** questi dispositivi tengono traccia delle attività quotidiane (e.g., numero di passi o distanza percorsa) così come il dispendio calorico e la frequenza del battito cardiaco. Questi dispositivi possono aiutare le persone a raggiungere obiettivi di salute come la perdita di peso o un regolare esercizio fisico monitorando le attività di interesse. A differenza

dei primi due tipi di dispositivi, che potrebbero richiedere una prescrizione medica per l'acquisto, questi ultimi possono essere acquistati senza di essa.

1.4.2 Dispositivi impiantabili

I dispositivi impiantabili o implantable medical devices (IMD) sono strumenti medici che costituiscono un grande progresso nel campo della medicina e rappresentano uno degli elementi principali nella sanità intelligente. Questi dispositivi vengono impiantati nel corpo del paziente, in parte o completamente, per periodi di tempo brevi o prolungati. Possono variare da dispositivi medici autonomi a dispositivi in grado di connettersi a una rete. Gli IMD possono aiutare in modo significativo a controllare e monitorare le condizioni del paziente per un'ampia gamma di malattie come diabete, Parkinson, aritmia cardiaca e così via. I pacemaker impiantati, le pompe per insulina, i defibrillatori e i neurostimolatori sono gli IMD più comunemente usati.

1.4.3 Dispositivi ambientali

Questi dispositivi sono disposti nell'area intorno al paziente con lo scopo di rilevare pattern nelle attività, qualità del sonno, visite al bagno, e avvisare gli operatori sanitari qualora si osservano pattern anomali. Tali sensori dovrebbero rendere le stanze più sicure e intelligenti per i pazienti con malattie croniche. I dispositivi ambientali includono [6]:

- sensori di movimento: rilevano i movimenti delle persone in una stanza;
- sensori per la temperatura ambientale: rilevano la temperatura dell'ambiente;
- sensori di pressione: monitorano il volume di gas, fluidi e aria;
- sensori nelle porte: rilevano lo stato della porta (ad es. aperta, chiusa) per i pazienti con Alzheimer o con malattie particolari;
- sensori di vibrazione: rilevano i movimenti del corpo dei pazienti sui letti;

- sensori di illuminazione: misurano automaticamente la luce naturale regolando l'illuminazione all'interno della stanza.

1.4.4 Dispositivi stazionari

Questa categoria include dispositivi che sono stazionari e di solito non con il paziente. Alcuni esempi:

- dispositivi di imaging: creano una rappresentazione visiva dell'interno di un organismo per l'analisi clinica e l'intervento medico, come l'imaging a risonanza magnetica (MRI), tomografia computerizzata (CT), scanner e raggi X;
- dispositivi chirurgici: strumenti di supporto utilizzati in interventi chirurgici o operazioni.

1.5 Applicazioni dell'IoMT

Il dominio sanitario è di grande importanza e l'IoT offre un ampio spettro di opportunità per renderlo migliore. Numerosi sensori e dispositivi medici possono connettersi alla rete offrendo accesso a importanti informazioni riguardo le condizioni dei pazienti. Queste informazioni possono essere utilizzate per diversi scopi come il monitoraggio dei pazienti a distanza, la previsione di malattie e la guarigione attraverso una maggiore comprensione dei sintomi e, in generale, il miglioramento del processo di diagnosi e trattamento tramite maggiore automazione e portabilità. I contesti applicativi in ambito sanitario possono essere suddivisi come mostrato in Figura 1.3. Di seguito verranno descritte alcune tra le applicazioni più importanti nell'ambito dell'IoMT [10].

- Disturbi cardiovascolari: le malattie cardiache sono una delle principali cause di mortalità nel mondo. Prevedere una malattia cardiaca è un compito complesso. Tuttavia, l'integrazione dell'IoT nei sistemi sanitari si è rivelato un modo straordinario per monitorare la salute dei pazienti e diagnosticare anomalie. I sensori più comunemente usati sono quelli per ECG, pressione

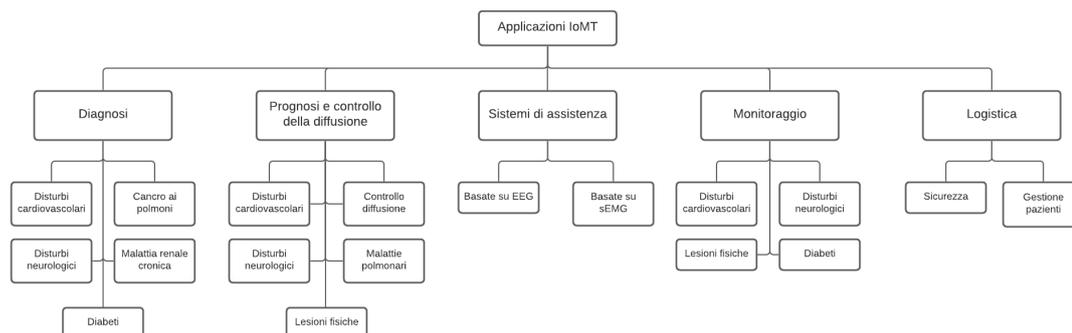


Figura 1.3: Classificazione applicazioni IoMT

sanguigna, battito cardiaco e temperatura corporea. I segnali ECG rappresentano l'attività del cuore a riposo. Possono essere usati per dedurre la frequenza cardiaca e il ritmo, e possono essere utili per diagnosticare l'allargamento del cuore a causa di pressione alta, frequenza cardiaca elevata, e aritmia o infarti. Inoltre, una volta identificati i principali fattori che portano a malattie cardiache, si può fare una previsione accurata della probabilità che una persona sviluppi tali malattie, prevenendo la stragrande maggioranza dei casi.

- **Cancro ai polmoni:** il cancro ai polmoni è il tumore più comune. In [26] viene presentato un modello per classificare le immagini polmonari ottenute con tomografia computerizzata per diagnosticare il cancro ai polmoni.
- **Disturbi neurologici:** per i disturbi neurologici come l'epilessia e l'Alzheimer, il monitoraggio basato su elettroencefalografia (EEG) è uno dei modi più importanti attraverso il quale sistemi sanitari intelligenti possono contribuire al monitoraggio dei pazienti. L'EEG è un metodo diagnostico utilizzato per rilevare i segnali elettrici nel cervello utilizzando elettrodi attaccati al cuoio capelluto. Un'altra tecnica che non è stata ampiamente utilizzata per la diagnosi neurologica è la magnetoencefalografia (MEG).
- **Diabete:** il diabete di tipo due è una delle malattie croniche più diffuse che può rivelarsi fatale se non tenuta sotto controllo. Un'efficiente identificazione di individui diabetici non diagnosticati porterà ad un trattamento migliore

e informato della malattia e ad un tasso di mortalità ridotto. In [18] viene proposto un modello di identificazione per pazienti non diagnosticati.

- **Malattia renale cronica:** si tratta di un problema di salute pubblica globale. In [25] viene effettuato uno studio della diagnosi della MRC attraverso algoritmi di Machine Learning.
- **Controllo della diffusione:** lo scenario mondiale attuale che vede l'incontrollabile diffusione della pandemia di coronavirus e le sue drastiche ripercussioni ha portato le persone a rendersi conto dell'impatto che tali malattie contagiose possono avere. Quindi, l'attenzione della ricerca è stata rivolta al controllo della diffusione di tali malattie contagiose. Ad esempio [19] propone l'identificazione di SARS-CoV-2 usando reti neurali.
- **Lesioni fisiche:** oltre il 10% delle morti nel mondo deriva da traumi causati da lesioni fisiche. Quando un infortunio fisico provoca molta perdita di sangue, l'apporto di ossigeno agli organi del corpo non è in grado di soddisfare la domanda. In una tale situazione si dice che il corpo si trova in shock emorragico. Più lo shock è grave maggiore sarà l'urgenza con la quale bisogna intervenire. In un tale scenario, una previsione accurata della gravità dello shock emorragico e del recupero del paziente è imperativo dare la priorità ai casi gravi, in modo da ridurre al minimo il tasso di mortalità.
- **Sistemi di monitoraggio:** il monitoraggio continuo della salute di una persona sta guadagnando molta rilevanza poiché non solo può ridurre drasticamente la mortalità derivante da emergenze improvvise, ma può anche promuovere la consapevolezza della propria salute e quindi portare a uno stile di vita più sano.
- **Sistemi di assistenza:** i sistemi di assistenza sono sistemi che fungono da strutture riabilitative o forniscono supporto nella vita quotidiana per le persone con disabilità. Questi costituiscono un'importante classe di applicazioni IoT basati su Machine Learning poiché molti di questi sistemi si affidano a capire se e in che misura la persona disabile è in grado di svolgere un determinato compito. A partire dal movimento degli arti (per il quale sistemi di

assistenza possono fornire riabilitazione fisica) così come essere in grado di ascoltare correttamente (mediante impianti cocleari).

- **Gestione pazienti:** i Pronto Soccorso negli ospedali spesso affrontano una situazione di sovraffollamento. I pazienti devono aspettare diverse ore prima che un letto diventi disponibile il che ha un impatto negativo sulla salute dei pazienti e sul tasso di mortalità globale. Se il numero di pazienti che devono essere ricoverati in ospedale potesse essere previsto utilizzando i registri ospedalieri, si potrebbe prevedere un eventuale sovraffollamento e gli ospedali sarebbero in grado di prevenirlo e migliorare la gestione dei pazienti stessi. Inoltre, in situazioni di crisi come nel caso di COVID-19, è importante stabilire le priorità dei pazienti sulla base della loro urgenza, rendendo il lavoro più efficiente. [8] ha proposto una soluzione per gestire la priorità dei pazienti attraverso un servizio basato su cloud.

1.6 Sfide e problemi nell'IoMT

Sebbene l'IoMT prometta connettività attraverso luoghi fisicamente distanti in cui pazienti, cliniche e ospedali potrebbero cooperare nei vari processi sanitari, ci sono diverse sfide che deve superare prima che possa affermarsi.

1.6.1 Gestione dei dati

Le sfide sulla gestione dei dati nell'IoMT sono simili a quelle affrontate dall'IoT negli altri domini. Tuttavia, i dati sanitari provengono da sensori medici attaccati agli umani per cui c'è un costante flusso di dati in arrivo ai nodi della rete. L'IoMT deve quindi gestire la complessità di questi dati in termini di varietà, volume e velocità. La varietà dei dati è una conseguenza dei vari formati legati all'eterogeneità dei dispositivi medici. Le sfide legate al volume e alla velocità dei dati sono più associate alle capacità hardware dei nodi di elaborare, archiviare e comunicare i dati stessi.

1.6.2 Scalabilità

Per costruire un sistema IoT su piccola scala bastano sensori per collezionare i dati e server attraverso i quali gli utenti possono accedere ai dati sanitari. Questa struttura può essere scalata ad un intero ospedale, così che i pazienti possono usufruire dei servizi sanitari, controllare aggiornamenti e monitorare il proprio stato di salute dal proprio cellulare. A sua volta, questo modello può essere scalato a un'intera città, se ci sono sensori ed antenne per raccogliere i dati, algoritmi ed API per processarli ed analizzarli, e interfacce intelligenti per informare gli utenti in tempo reale. In una smart city, tutti i dati possono essere raccolti, elaborati e analizzati dagli smartphone e il feedback sarà inviato ai pazienti per consentire loro di conoscere il proprio stato di salute e i risultati delle loro visite mediche. Ciò consentirebbe di migliorare l'efficienza, risparmiando tempo di attesa e costruendo un rapporto diretto e di fiducia tra il personale medico e i pazienti.

1.6.3 Design interfaccia utente

Uno dei fattori chiave dell'IOMT è l'interfaccia delle tecnologie front-end come sensori, smartphone, tablet, computer ecc. Quando i dispositivi medici IoT vengono progettati e forniti ai pazienti che non hanno esperienza nell'uso di strumenti tecnologici, diventa molto importante che gli utenti finali possano comprenderne autonomamente l'uso. In generale, gli utenti finali hanno una conoscenza limitata della rete wireless, della sincronizzazione dei sensori e di altre funzionalità del dispositivo. Inoltre, quando i dispositivi andranno in ambienti remoti, la configurazione dei sistemi di IOMT dovrà essere semplice e autonoma. Ad esempio, la popolazione anziana sarà una delle maggiori parti interessate dall'IOMT. Pertanto, le interfacce dei dispositivi devono essere di facile utilizzo per l'utente (paziente) e richiedere il minimo coinvolgimento di esperti. Una delle pratiche di successo è il design partecipativo, in cui le parti interessate o gli utenti finali del dispositivo IOMT potrebbero diventare parte del team di progettazione per fornire un feedback continuo.

1.6.4 Sicurezza e privacy

Ogni singolo dispositivo IoT potrebbe presentare un rischio potenziale che potrebbe essere sfruttato per danneggiare l'utente finale o comprometterne la privacy. Se la sicurezza è compromessa, è possibile accedere alle informazioni degli utenti finali tramite autenticazioni non autorizzate. Inoltre, ciò può comportare rischi per l'incolumità degli individui. La sicurezza e la privacy nell'IoMT si estendono all'intero ciclo di vita del sistema a partire dalla generazione delle specifiche, all'implementazione e alla distribuzione. Tuttavia, proteggere l'ecosistema IoMT è un compito sofisticato e impegnativo.

1.6.5 Interoperabilità e standardizzazione

In generale, l'IoT ha sollevato preoccupazioni nell'area della standardizzazione. Tutti i produttori, i fornitori di servizi e gli utenti finali cercano standard di operatività sia all'interno che tra il dominio avviato dalle applicazioni IoT. La complessità della standardizzazione risiede nel fatto che l'IoT mira a catturare un'ampia gamma di discipline che sono, in generale, regolate da diversi affari normativi. Nel caso dell'IoMT, la complessità aumenta anche a causa delle rigide normative imposte dagli standard medici. I problemi di interoperabilità verranno approfonditi nel corso dei capitoli successivi proponendo una soluzione per il caso di studio in esame.

Capitolo 2

Interoperabilità ecosistema IoMT

Come emerso dal primo capitolo, nel settore sanitario vengono utilizzati diversi dispositivi IoT. Questo determina la creazione di un grande volume di dati eterogenei, i quali possono essere combinati con quelli presenti nelle cartelle elettroniche dei pazienti (EPR) e con le informazioni all'interno di grandi database. Diverse fonti di dati possono essere rilevanti nell'erogazione di servizi di sanità elettronica di alta qualità, e.g., dati genomici, EPR, dati dei dispositivi IoT e dati clinici. Eseguendo analisi sui dati aggregati, il personale medico è in grado di fornire servizi migliori. Tuttavia, come anticipato, l'interoperabilità deve essere garantita per analizzare i dati in tempo reale e fornire servizi di sanità elettronica di alta qualità. Per questo motivo, l'interoperabilità rappresenta al giorno d'oggi una delle sfide più importanti dell'ecosistema IoMT. L'Healthcare Information and Management Systems Society (HIMSS) definisce l'interoperabilità nell'assistenza sanitaria come “la capacità di diverse tecnologie dell'informazione, sistemi e applicazioni software di comunicare, scambiare dati e utilizzare le informazioni che sono state scambiate”, ponendo enfasi sul fatto che lo schema e gli standard per lo scambio dei dati dovrebbero consentirne la condivisione [12]. Per affrontare questa sfida, sono emerse diverse iniziative di standardizzazione che verranno descritte in questo capitolo.

2.1 Integrating the Healthcare Enterprise

IHE è un'organizzazione senza scopo di lucro fondata nel 1998 promossa dagli operatori sanitari e dall'industria, che mira a migliorare il modo in cui i dispositivi condividono informazioni nel settore sanitario. Il suo obiettivo principale non è sviluppare nuovi standard tecnici, ma piuttosto formulare linee guida per l'uso di standard esistenti sulla base di scenari applicativi chiaramente delineati. Promuove l'uso di standard come DICOM e HL7 per affrontare specifiche esigenze cliniche. IHE raccoglie i requisiti del singolo caso, identifica gli standard disponibili e sviluppa linee guida tecniche per i produttori attraverso i cosiddetti profili. È possibile fare una distinzione approssimativa tra profili di integrazione e profili di contenuto. I profili di integrazione descrivono gli attori (ruoli) che effettuano transazioni, scambiando informazioni tra di loro e garantendo, quindi, l'interoperabilità sintattica. L'interoperabilità semantica si ottiene attraverso profili di contenuto in cui viene dedicata una struttura per ogni elemento di dati. IHE lavora nei cosiddetti domini, che corrispondono principalmente ai campi medici (ad esempio radiologia, oftalmologia o cardiologia). Il dominio dell'infrastruttura informatica include profili di integrazione condivisi da tutti i domini [9].

2.2 ISO/IEEE 11073

Gli standard ISO/IEEE 11073, a cui si fa riferimento anche come standard X73, sono il risultato di una collaborazione tra International Organization of Standardization (ISO) e IEEE. Questi standard sono stati stabiliti nel momento di culmine della loro necessità nel dominio sanitario. Essi mirano a garantire la comunicazione tra dispositivi medici e sistemi informatici. Gli obiettivi primari sono [5]:

- Fornire interoperabilità plug-and-play in tempo reale per i dispositivi medici collegati al paziente. In tempo reale significa che i dati da più dispositivi possono essere recuperati, associati ad una marcatura temporale e visualizzati o elaborati in frazioni di secondo.
- Facilitare lo scambio in modo efficiente di dati relativi al paziente e ai dispositivi medici, acquisiti presso i punti di cura, in tutti gli ambienti sanitari.

Ciò significa che le informazioni acquisite presso il PoC (ad es. dati sui parametri vitali del paziente) possono essere archiviati, recuperati ed elaborati da molti diversi tipi di applicazioni senza un supporto completo di software e apparecchiature e senza inutili perdite di informazioni. Questo standard è particolarmente mirato ai Point of Care Devices (PoCD), come monitor dei parametri vitali del paziente, pompe infusionali, dispositivi ECG, ecc.

2.2.1 ISO/IEEE 11073 PHD

ISO/IEEE 11073 PHD definisce standard di comunicazione per lo scambio di informazioni nel contesto di Personal Health Devices (PHD), ovvero dispositivi sanitari che vengono mantenuti dal paziente. Siccome ISO/IEEE 10073 non supporta il protocollo IP, è difficile fornire servizi sanitari in ambienti IoT. Sotto questo insieme di standard, ciascun sistema comprende tre modelli principali: Domain Information Model (DIM) per la rappresentazione dei dati, Service Model (SM) per definire l'accessibilità dei dati e Communication Model (CM) per la comunicazione dei dati.

2.2.2 ISO/IEEE 11073 SDC

Gli standard ISO/IEEE 11073 SDC seguono il paradigma Service Oriented Architecture (SOA). Le specifiche mirano a raggiungere l'interoperabilità tra dispositivi medici. Essi includono:

- IEEE 11073-20702: per standard dei servizi Web con una comunicazione per supportare lo scambio di dati elettronici sanitari e controllo dei dispositivi medici presso il punto di assistenza (PoC) all'interno della rete.
- IEEE 11073-10207: DIM e SM per la comunicazione service-oriented tra dispositivi PoC.
- IEEE 11073-20701: definisce un'architettura service-oriented per dispositivi medici PoC e sistemi di informazione.

2.3 Health Level Seven International

HL7 è un'organizzazione non a scopo di lucro che sviluppa standard. Essa mira a fornire un framework completo e i relativi standard per lo scambio, integrazione, condivisione e recupero delle informazioni sanitarie elettroniche a supporto dei servizi di pratiche cliniche, gestione, consegna e valutazione della salute. HL7 è responsabile dell'interoperabilità sia semantica che sintattica. Gli standard semantici si riferiscono ai vocabolari utilizzati per denotare i dati, come eventi, risultati e condizioni. Gli standard sintattici si riferiscono a come il sistema del mittente impacchetta tali dati prima dell'invio. Poiché HL7 si evolve continuamente con l'uso e con l'esperienza, al giorno d'oggi ne sono state rilasciate diverse versioni suddivisibili in HL7 v2 e HL7 v3.

2.3.1 HL7 v2

La versione 2.x (v2) di HL7 è uno standard basato su messaggi ed è il cavallo di battaglia per quanto riguarda lo scambio di dati elettronici nel dominio clinico e probabilmente lo standard più utilizzato al mondo per l'assistenza sanitaria. Questo standard di messaggistica consente lo scambio di dati clinici tra i sistemi. È progettato per supportare sia un sistema centralizzato di assistenza al paziente che un sistema più distribuito. Nonostante la versione 2 di HL7 sia stata rilasciata decenni fa è ancora in fase di aggiornamento e miglioramento.

I primi messaggi tra sistemi tendevano ad essere inviati e ricevuti all'interno della stessa organizzazione. Lo standard si è evoluto per accrescimento; cioè, una volta identificato, un nuovo campo veniva aggiunto alla parte già esistente. Man mano che la comunità HL7 è cresciuta e la tecnologia è progredita, si è anche evoluto un modello di dati meglio organizzato.

Un messaggio HL7 v2 viene identificato da una stringa di tre caratteri, per esempio, "ACK". Ciò che scaturisce l'invio di un messaggio prende il nome di "trigger" ed è in genere un evento, come il trasferimento di un paziente da una struttura all'altra o un paziente che riceve una cura. Un messaggio v2 non ha dimensione e contenuto fissi, permettendo al messaggio di espandersi o contrarsi a seconda dei dati richiesti.

Un messaggio HL7 è costituito da uno o più segmenti visualizzati su righe di testo diverse. Ogni segmento è costituito da uno o più campi separati dal carattere "|". Se un campo contiene altri campi, questi sono normalmente separati da caratteri "^".

Siccome i messaggi HL7 vengono utilizzati per comunicare tutti i tipi di informazioni di assistenza sanitaria, a volte devono contenere dati personalizzati che non possono essere inclusi in alcun segmento definito per il tipo di messaggio. A tale scopo, lo standard HL7 consente di creare un segmento Z con campi personalizzati per trasmettere questi dati. Per convenzione, tutti i segmenti personalizzati iniziano con la lettera Z. I segmenti Z possono essere posizionati ovunque in un messaggio HL7, tuttavia in genere si trovano come ultimo segmento. Le applicazioni che elaborano i messaggi HL7 sono generalmente configurate per ignorare i segmenti Z HL7 che non sanno come gestire.

Listato 2.1: Esempio messaggio HL7 v2

```

1 MSH|^~\&|EPIC||SMS|SMSDT|201501011408||ADT^A04|9000123|D
  |2.7|
2 EVN|A04|201501011408
3 PID||0493575^^^2^ID 1|454721||DOE^JOHN
4 NK1||CONROY^MARI|SPO|||(216)731-4359
5 NK1||DOE^JOHNNY^^^^|CHD|||(216)731-4222
6 NK1||DOE^ROBERT  ^^^^|CHD|||(216)731-4222
7 PV1||0|168 ~219~C~P

```

2.3.2 HL7 v3

Alla fine del 2005, la comunità HL7 ha rilasciato la prima versione di HL7 v3, la Normative Edition 2005. A metà del 2006 è stata invece pubblicata la Normative Edition 2006. Con queste due versioni iniziali, molte sezioni dello standard sono state completate, ma diverse aree significative verranno definite o create solo nelle versioni successive. Il valore principale del nuovo standard consiste in un modello di dati esplicito, definizioni chiare e più casi d'uso consentendo meno flessibilità nei

singoli elementi del messaggio. Infatti, uno standard più rigido promette interfacce più facili per gli utenti.

HL7 v3 mira a supportare i flussi di lavoro dell'assistenza sanitaria. Il framework di sviluppo HL7 v3 (HDF) documenta il livello di messaggistica, processi, regole e artefatti. L'HL7 Reference Information Model (RIM) esprime il contenuto dei dati richiesti in uno specifico contesto clinico o amministrativo, affrontando la semantica dei messaggi basati sul formato XML.

2.4 HL7 FHIR

2.4.1 Sviluppo di FHIR

Nel 2011, a causa delle critiche e dei feedback sulla v3, HL7 ha organizzato un gruppo con l'obiettivo di sviluppare uno standard in risposta alle esigenze del mercato di metodi più veloci, più facili e migliori per scambiare la quantità di dati sanitari in rapida crescita. Di conseguenza, è stato redatto FHIR usufruendo delle lezioni apprese dagli standard precedenti. Mentre HL7 v2 e v3 erano orientati ai messaggi, il paradigma architetturale alla base di FHIR è Representational State Transfer (REST). REST è un'architettura software per lo sviluppo di sistemi web alla base dell'Internet moderno. In un sistema REST, i client effettuano richieste a un server utilizzando l'Hypertext Transfer Protocol (HTTP), che si basa sui protocolli TCP/IP. Le architetture RESTful portano a interfacce leggere che consentono una trasmissione più veloce ed elaborazione di strutture dati, più adatte a dispositivi mobili come cellulari e tablet.

Dato che REST è un paradigma architetturale integrale su Internet, e dato che le interfacce REST sono leggere, lo sviluppo dell'implementazione di FHIR si presta a metodi di sviluppo agile. Le risorse FHIR, o le entità utilizzate per modellare i dati per lo scambio di informazioni, sono sviluppate in modo iterativo con fine ultimo la flessibilità.

2.4.2 Cos'è FHIR

FHIR è costruito attorno alle risorse, ovvero piccoli componenti modulari che rappresentano le informazioni cliniche in un formato orientato agli oggetti. A causa di nuovi casi d'uso, ad ogni rilascio di FHIR vengono introdotte nuove risorse. Ogni tipo di risorsa è una rigida gerarchia di elementi, ognuno dei quali può essere un valore primitivo, un valore complesso o un insieme di elementi figli. In aggiunta, un elemento può essere una chiave esterna, ossia un riferimento ad una risorsa diversa. La specifica FHIR fornisce una definizione logica, una definizione UML e rappresentazioni attraverso modelli XML e JSON per ciascuna risorsa.

Le pubblicazioni di FHIR si suddividono nel seguente modo:

- DSTU1: le prime pubblicazioni di FHIR sono Draft Standard for Trial Use 1, introdotte per la prima volta alla fine del 2012.
- DSTU2: Draft Standard for Trial Use 2, sviluppate tra il 2014 e il 2015.
- STU3: Standard for Trial Use 3, sviluppata tra il 2015 e il 2017.
- R4: la versione più moderna è la R4, con la sua ultima versione datata dicembre 2018. Questa specifica include oltre 100 risorse.

Di seguito la struttura di una risorsa base FHIR in formato JSON:

Listato 2.2: Esempio risorsa base FHIR

```
1 {
2   "resourceType" : "[name]",
3   "id" : "<id>", // Logical id of this artifact
4   "meta" : { Meta }, // Metadata about the resource
5   "implicitRules" : "<uri>", // A set of rules under
   which this content was created
6   "language" : "<code>" // Language of the resource
   content
7 }
```

2.4.3 FHIR come API

Dato che REST è stato utilizzato come paradigma architetturale, la descrizione di FHIR include le specifiche per delle API. Fedele alla sua promessa di fare affidamento su standard Internet aperti, FHIR specifica il protocollo web HTTP per lo scambio di dati. Questa decisione progettuale prevede una distinzione tra server e client. Mentre i client conformi a FHIR forniscono applicazioni leggere che usufruiscono di dati FHIR, i server FHIR si occupano di sicurezza, threading, ricerca, indicizzazione e persistenza. Sia i client che i server conformi a FHIR rappresentano i dati mediante risorse FHIR e scambiano dati secondo le specifiche API.

Le API FHIR si integrano perfettamente con le risorse. Poiché ogni risorsa ha un identificatore univoco, ogni risorsa ha un URL univoco attraverso il quale può essere recuperata dal server FHIR utilizzando il metodo GET. La funzionalità di ricerca è centrale in quanto le richieste GET possono essere create specificando parametri da confrontare con valori di elementi specifici delle risorse. Ad esempio, è possibile interrogare un server FHIR per trovare un paziente con uno specifico ID, per trovare pazienti con date di nascita in un intervallo specifico e altro ancora.

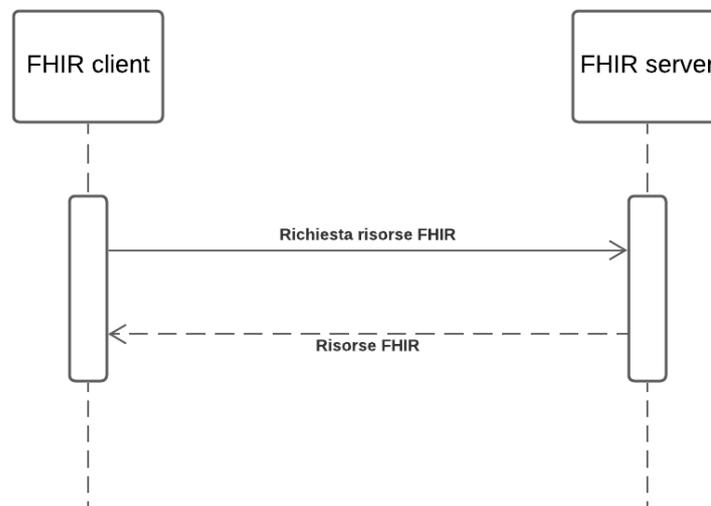


Figura 2.1: Diagramma di sequenza FHIR API

Nella realtà dei fatti, tuttavia, le implementazioni FHIR presso vari fornitori

di cure non rispecchiano una semplice architettura client-server. Questo perché gli operatori sanitari dispongono di sistemi radicati e di lunga data. Sebbene sarebbe ideale per tutte le implementazioni utilizzare FHIR come "lingua franca", magari costruendo una nuova implementazione da zero, ciò è difficile da realizzare per la maggior parte delle infrastrutture mediche. Di conseguenza, le istituzioni spesso usano FHIR con due strategie principali:

- Protocol Middleware: sviluppo di middleware FHIR che funge da interfaccia tra le operazioni FHIR in entrata e i protocolli di sistema interni.
- Data Mapping Module: sviluppo di un modulo che gestisca una mappatura bidirezionale tra i modelli informativi interni e le risorse FHIR.

2.4.4 Estensioni e profili

Uno dei motivi principali dell'estensibilità di FHIR consiste nella variabilità dell'assistenza clinica all'interno di ogni paese e nel mondo. Infatti, non solo gli standard divergono all'interno di un singolo paese, ma regolamentazioni e finanziamenti diversi precludono pratiche di cura comuni. Comprese queste problematiche e dato il fallimento del rigido standard v3, HL7 aveva bisogno di garantire spazio per flessibilità ed estensibilità, che spesso sono in contrasto con la facilità d'uso.

Nelle versioni recenti, FHIR ha rilasciato un costrutto che consente alle implementazioni di specificare vincoli o estensioni alle risorse generiche. Le implementazioni possono profilare una risorsa base utilizzando un'utile risorsa chiamata StructureDefinition. Solitamente la profilazione di una risorsa viene effettuata per vincolare ulteriormente i limiti di cardinalità di uno specifico elemento. In alternativa, un elemento che è definito come facoltativo può essere modificato per essere reso obbligatorio.

In aggiunta, le implementazioni possono estendere una risorsa base definendo una StructureDefinition che caratterizza ulteriori elementi dell'oggetto. Per questo motivo, ogni risorsa FHIR ha un elemento opzionale chiamato Extension, che può essere incluso un numero illimitato di volte. Ogni elemento di estensione ha un URL che riporta alla definizione dell'implementazione.

Offrire semplicemente strumenti di flessibilità ed estensibilità non è sufficiente per garantire interoperabilità. Oltre a questi strumenti, la specifica FHIR impone che le implementazioni esponano un endpoint per una dichiarazione di conformità. La risposta restituita da questo endpoint spiega le operazioni e le risorse supportate dal server. Inoltre, le estensioni e i profili per le risorse base sono indicati nella dichiarazione di conformità. In breve, la dichiarazione di conformità di un server FHIR descrive le capacità e le personalizzazioni dell'implementazione.

HL7 presenta anche il proprio registro di implementazioni FHIR. Oltre a un registro di tutte le implementazioni FHIR, che gli sviluppatori propongono di propria iniziativa, il registro include archivi di profili ed estensioni sviluppati durante varie implementazioni. Questa pratica consente agli sviluppatori di utilizzare estensioni già definite per risorse, migliorando la coerenza all'interno dello standard e riducendo il carico di sviluppo.

2.4.5 Standard terminologici: LOINC e SNOMED

Gli standard terminologici sono considerati un requisito indispensabile per l'interoperabilità semantica. Forniscono identificatori universali e assegnabili in modo univoco che consentono lo scambio di dati clinici tra sistemi informatici eterogenei.

La relazione tra la struttura e la semantica dei dati in FHIR è integrata all'interno delle risorse e quindi deve essere affrontata durante la modellazione delle informazioni dei profili. Di conseguenza, una parte importante della profilazione delle risorse FHIR è collegare i suoi elementi ai sistemi terminologici e ai set di valori più appropriati. I due sistemi di maggior interesse sono:

- **LOINC**: è un database e un linguaggio comune (insieme di identificatori, nomi e codici) per identificare misurazioni, osservazioni e documenti sanitari.
- **SNOMED CT**: è una raccolta sistematicamente organizzata di termini medici che fornisce codici, termini, sinonimi e definizioni utilizzati nella documentazione clinica. È considerato lo standard terminologico più completo e complesso nel campo medico.

2.5 Altri standard

2.5.1 Continua Health Alliance

Continua Health Alliance (ora conosciuta come PCHA) è stata fondata nel 2006, comprende rappresentanti del settore sanitario e dei dispositivi medici e si concentra sul miglioramento dell'assistenza sanitaria in tutto il mondo proponendo soluzioni di interoperabilità. Inoltre, affronta ostacoli tecnici, normativi e finanziari sulla telemedicina. PCHA mira ad allineare diverse iniziative, standard, fornitori e domini, concentrandosi sulla gestione della malattia, l'invecchiamento in modo indipendente e la salute verso l'ottenimento di un ecosistema interoperabile. Le Continua Design Guidelines forniscono specifiche per un framework verso soluzioni sanitarie che garantiscono uno scambio di dati interoperabile e sicuro in armonia con gli standard e i protocolli sanitari esistenti.

2.5.2 oneM2M

Diverse organizzazioni hanno lavorato allo sviluppo di standard per risolvere la frammentazione del panorama IoT. Tra questi, l'iniziativa oneM2M mira a sviluppare specifiche tecniche standardizzate considerando una piattaforma orizzontale comune. Lo standard oneM2M si concentra sull'interoperabilità nei livelli di servizio piuttosto che nei livelli di rete e Internet. Di conseguenza, lo standard oneM2M definisce funzioni di servizio comuni per i sistemi IoT, tra cui registrazione e rilevamento dei dispositivi, gestione dei dati, sottoscrizione e notifica, semantica, sicurezza e altri [28].

Inizialmente, lo standard IoT oneM2M non supportava i servizi sanitari forniti dai dispositivi indossabili che utilizzavano ZigBee e Bluetooth. Tuttavia, sviluppi recenti hanno abilitato il funzionamento di dispositivi non conformi a oneM2M con la piattaforma lato server oneM2M.

2.5.3 ZigBee Health Care

ZigBee Health Care è promosso dalla ZigBee Alliance, che supporta ZigBee. ZigBee Health Care fornisce uno standard per lo scambio di dati tra dispositivi medici

e non medici; supporta ISO/IEEE 11073 per la comunicazione di dispositivi PoC. I profili applicativi pubblici di ZigBee consistono in standard per messaggi, formati di messaggi e azioni di elaborazione, consentendo la creazione di applicazioni distribuite interoperabili.

2.6 Point-of-Care Device Implementation Guide

Una "Implementation Guide" (IG) è un insieme di regole su come le risorse FHIR vengono utilizzate (o dovrebbero essere utilizzate) per risolvere un particolare problema, con annessa documentazione per supportare e chiarirne l'utilizzo. In genere, le FHIR IG vengono pubblicate sul Web dopo essere state generate utilizzando il FHIR Implementation Guide Publisher [1].

In particolare, in questa sezione verrà trattata l'IG relativa ai Point-of-Care Device, ovvero quei dispositivi utilizzati per ottenere risultati diagnostici mentre sono con il paziente o vicino ad esso. Utilizzati negli studi medici e negli ospedali, i dispositivi PoC forniscono un rapido feedback su molti tipi di test medici.

Questa IG definisce l'uso delle risorse FHIR per trasmettere misurazioni e dati di supporto dai dispositivi medici PoC ai sistemi di ricezione per cartelle cliniche elettroniche, supporto decisionale clinico e archiviazione di dati medici per scopi di misurazione e ricerca. La PoCD Implementation Guide è stata sviluppata dall'iniziativa Devices on FHIR (DoF). Si tratta di uno sforzo collaborativo tra HL7 Health Care Devices Working Group, Integrating the Healthcare Enterprise (IHE) e Personal Connected Health Alliance (PCHAlliance) / Continua. L'obiettivo del gruppo DoF è garantire un uso coerente di FHIR per lo scambio di informazioni sui dispositivi sanitari, tra cui [3]:

- massimo allineamento tra PoCD e Personal Health Devices (PHD);
- allineamento con le specifiche basate su HL7 v2 a partire dal framework IHE Patient Care Device (PCD);
- mappatura coerente della terminologia e dei modelli informativi ISO/IEEE 11073.

2.6.1 Panoramica modello generale

La visione più semplice prende il nome di "modello piatto" (flat model), è rappresentata semplicemente da una risorsa Observation che punta a un singolo dispositivo come fonte dati. Questo modello è sufficiente per l'uso clinico quando si desidera enfatizzare la cura del paziente di ora in ora e minuto per minuto, senza sovraccaricare l'utente con altre complessità. Per altri casi d'uso, può essere importante rappresentare un contesto più ampio su come è stata effettuata la misurazione.

La spina dorsale per rappresentare un dispositivo in questa IG è il modello generico dell'IEEE 11073 Medical Device Communications Domain Information Model. Si tratta di un modello standard ampiamente utilizzato, basato su una rappresentazione gerarchica di un dispositivo medico come un sistema complessivo con sottosistemi logicamente distinti, che a loro volta hanno livelli aggiuntivi contenenti oggetti.

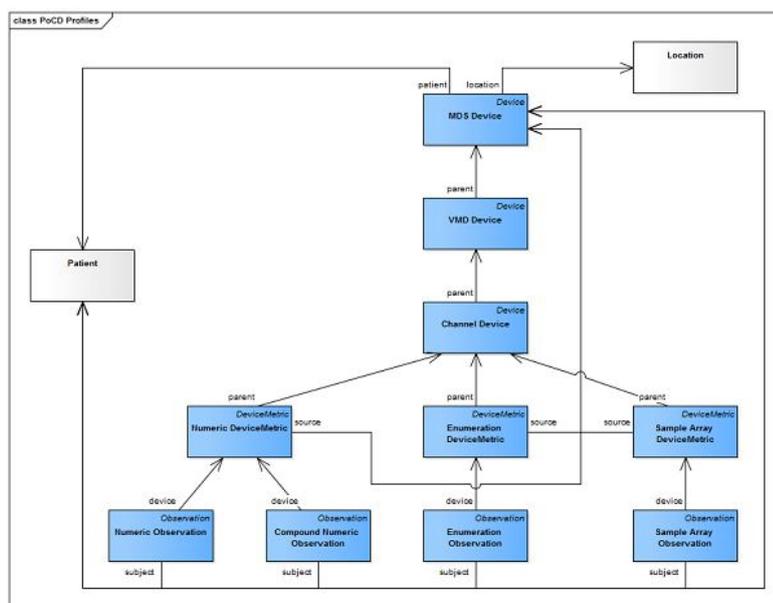


Figura 2.2: Modello PoCD [3]

La radice dell'albero è il sistema complessivo, ed è identificato nel modello come Medical Device System (risorsa MDS) che rappresenta l'intero dispositivo, con sottosistemi logici identificati come Virtual Medical Devices (VMD). Questi

sottosistemi logici VMD possono anche avere un aspetto fisico: possono essere rimovibili, come in un modulo di misurazione in un monitor a parametri vitali, quindi avere un'identità individuale. È possibile che i VMD debbano avere le misurazioni raggruppate in canali logici. A volte non è necessario modellarlo, ma in casi come quelli di pompe per infusione o moduli EEG, esiste una partizione significativa dei dati in canali che possono essere fondamentali per il trattamento sicuro del paziente.

Al livello più basso, le osservazioni corrispondono a ciò che il Domain Information Model IEEE 11073-10101 chiama "metric". Queste non sono necessariamente singole misurazioni quantitative. Possono essere [3]:

- enumerazioni, ovvero variabili qualitative o categoriali, come una valutazione "lieve - moderato - grave".
- insieme di valori strettamente correlati che è meglio tenere insieme e registrare come valore composto.
- un vettore di quantità.

Capitolo 3

Caso di studio: monitor parametri vitali

In questo capitolo verrà fornito inizialmente un quadro generale sul dispositivo preso in esame come caso di studio ovvero il monitor a parametri vitali. Per far ciò, si partirà dando una definizione del monitor per poi soffermarsi sul modello adoperato dall'ospedale Bufalini di Cesena, l'Infinity M540 - Draeger. Quindi, si analizzeranno i dispositivi concorrenti nel settore in commercio, ponendo l'accento sull'aspetto dell'interoperabilità e della possibile integrazione in un contesto di ospedale 4.0. Si conclude con una proposta di modellazione del dispositivo monitor parametri vitali prima mediante le risorse FHIR base e poi seguendo la PoCD Implementation Guide.

3.1 Stato dell'arte

3.1.1 Cos'è un monitor parametri vitali

Il monitoraggio dei parametri vitali è fondamentale per riconoscere lo stato fisico del paziente in un determinato momento. È un indicatore di come la persona mantiene l'equilibrio fisiologico e di come reagisce o progredisce nella propria malattia. Le osservazioni fisiologiche o il monitoraggio dei segni vitali sono un principio fondamentale della pratica infermieristica all'interno di un reparto di terapia intensiva

in cui il flusso di pazienti è costante e l'esito delle decisioni dipende dal tempo in cui le stesse vengono prese. Tradizionalmente, i segni vitali comprendevano quattro misurazioni, temperatura corporea, polso o frequenza cardiaca, frequenza respiratoria e la pressione arteriosa, con la saturazione sanguigna aggiunta nell'ultimo decennio diventando così il quinto parametro vitale accettato [13].

In generale, il monitor dei parametri vitali può essere definito come un dispositivo medico che è in grado di acquisire una serie di importanti parametri sulla salute del paziente, visualizzati in forma grafica e numerica. Tale dispositivo medico è anche noto come monitor multi parametrico. I segnali visualizzati sul monitor multi parametrico includono ECG, frequenza respiratoria, pressione sanguigna (BP), saturazione sanguigna (SpO₂) e temperatura corporea, come nell'applicazione d'esempio mostrata nella Figura 3.1. In questo dispositivo, tutti i segnali vengono visualizzati contemporaneamente sia in forma grafica che in forma numerica per un'ulteriore analisi da parte degli esperti medici [17].



Figura 3.1: Esempio di applicazione monitor a parametri vitali [17]

3.1.2 Infinity M540

Di seguito verranno descritte le caratteristiche ed il funzionamento del monitor a disposizione presso l'ospedale Bufalini di Cesena, l'Infinity M540. Anche semplicemente chiamato M540, è concepito per il monitoraggio di informazioni fisiologiche multi parametriche sul paziente ottenute da hardware connesso in ambienti in cui l'assistenza al paziente è fornita da operatori sanitari qualificati. L'M540 è progettato per monitorare un paziente alla volta. L'M540 è in grado di monitorare più di venti parametri, compresi quelli standard precedentemente descritti.

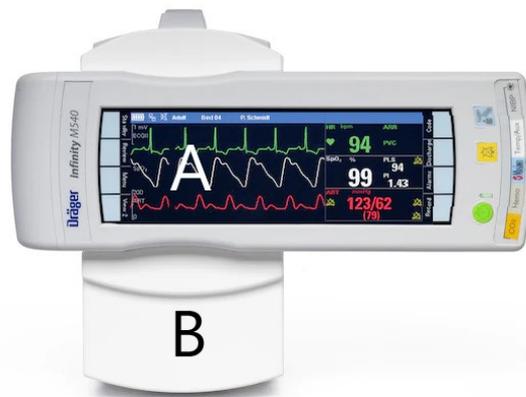


Figura 3.2: A monitor M540, B stazione di aggancio M500

Quando l'M540 è agganciato all'Infinity M500 e fa parte di una configurazione Infinity Acure Care System (IACS), esso diventa un modulo incaricato dell'acquisizione dei parametri vitali e dell'elaborazione dei dati per Infinity C500/700. L'M540 può essere utilizzato anche come monitor autonomo collegato a un Infinity M500. L'M540 fornisce un continuo monitoraggio del paziente anche quando viene sganciato dall'M500 per il trasporto del paziente stesso, infatti è dotato di un'opzione wireless che gli consente di trasmettere i dati all'ICS (Infinity Central Station). L'M540 agganciato a un M500 può essere configurato come stand-alone per caricare la batteria quando non fa parte di una configurazione IACS.

Dal punto di vista dell'interoperabilità i dispositivi della Draeger si interfacciano a un Infinity Gateway Server, il quale facilita lo scambio di importanti informa-

zioni cliniche tra la rete Infinity e i sistemi ospedalieri esistenti. Esso è progettato per fornire agli sviluppatori la massima flessibilità, utilizzando protocolli sanitari e formati di dati standard per la gestione delle comunicazioni tra i sistemi più disparati.

In particolare, tutte le interfacce Infinity utilizzano HL7 versione 2.3, se non diversamente specificato, implementato su socket TCP/IP orientate alla connessione e MLLP come protocollo a livello inferiore, sebbene alcune delle interfacce accettano anche altre versioni.

Due sistemi sono sempre coinvolti in un'interfaccia HL7. Un sistema è il master e avvia la connessione mentre l'altro è lo slave e accetta la connessione. Ogni interfaccia Infinity specifica se il server Draeger HL7 sarà il master o lo slave della connessione. Tipicamente per le interfacce "unsolecited", il master è la fonte dei dati da trasferire, al contrario, per un'interfaccia "solecited", lo slave è la fonte di i dati.

3.1.3 Analisi dispositivi in commercio

Dopo aver descritto il monitor a parametri vitali in utilizzo presso l'ospedale Bufalini, si vuole dare uno sguardo più ampio agli altri modelli in commercio. In particolare si vuole fare una panoramica cercando di individuare quei modelli che vanno maggiormente in una direzione IoT e quindi sarebbero più facilmente integrati in un contesto di ospedale 4.0. Tra i dispositivi in tal senso più rilevanti troviamo:

- SunTech CT50: questo dispositivo implementa diversi profili di comunicazione di IHE e ha la possibilità di trasmettere i dati di misurazione dal dispositivo in conformità con i protocolli di messaggistica HL7.
- mindray VS-900: questo modello consente una trasmissione wireless intelligente senza operazioni aggiuntive, avendo una rete Wi-Fi integrata. È in grado di effettuare una sincronizzazione in tempo reale con la stazione centrale proprietaria. Dal punto di vista dell'interoperabilità è conforme allo standard HL7 e ai profili IHE, consentendo quindi di trasmettere i dati dal monitor ad una eventuale cartella clinica elettronica.

- Riester RVS-100: L'RVS-100 può comunicare tramite una connessione cablata o wireless con il sistema informatico dell'ospedale secondo lo standard HL7.
- NORTHERN Aquarius: anche questo dispositivo adotta il protocollo HL7 per connettersi con il sistema informatico ospedaliero.

Da questa breve panoramica si può notare come molti monitor in commercio, compreso quello dell'ospedale Bufalini, adottino lo standard di comunicazione HL7 per lo scambio dei messaggi. Tuttavia, la maggior parte sfrutta software della casa produttrice che ne ostacolano l'integrazione in un contesto ospedaliero IoT. Sulle orme di questa problematica, nella prossima sezione si propone una modellazione con risorse FHIR, con l'obiettivo di poter integrare al sistema informatico ospedaliero qualsiasi modello di monitor parametri vitali.

3.2 Modellazione con FHIR

Prima di procedere con la modellazione seguendo le linee guida proposte dalla PoCD Implementation Guide, è utile introdurre le risorse base di FHIR che possono essere utilizzate in una prima visione del dispositivo monitor parametri vitali e delle rispettive misurazioni. In quest'ottica, sono d'interesse tre principali risorse: Device, DeviceMetric e Observation.

3.2.1 Device

In FHIR la risorsa Device fa riferimento a un tipo di prodotto, medico o non medico, che viene utilizzato per fornire assistenza sanitaria. Questa risorsa è di carattere amministrativo e tiene traccia delle singole istanze di un dispositivo e della sua posizione. La risorsa Device è referenziata da altre risorse, ad esempio per registrare quale dispositivo ha eseguito un'azione come una procedura o un'osservazione, oppure quando si prescrivono e dispensano dispositivi per l'uso del paziente o per ordinare rifornimenti. Infine viene utilizzata per registrare e trasmettere informazioni sull'identificatore univoco del dispositivo (UDI) relativo ad

un dispositivo come l'impianto di un paziente. Un importante aspetto da sottolineare è che in questa rappresentazione il dispositivo viene descritto come elemento amministrativo (ad esempio contiene le informazioni del produttore), quindi non cambia molto nel tempo. Al contrario, la risorsa DeviceMetric descritta in seguito modella la parte fisica, incluso lo stato operativo ed è molto più volatile.

3.2.2 DeviceMetric

La risorsa DeviceMetric descrive le proprietà statiche obbligatorie che caratterizzano una misurazione, un'impostazione o un calcolo del segnale biologico diretto o derivato, quantitativo o qualitativo, prodotto da un dispositivo medico. La risorsa DeviceMetric può essere utilizzata anche per descrivere le proprietà non statiche ma altamente rilevanti per la metrica come lo stato della metrica, l'ora e il tipo dell'ultima calibrazione della metrica, la modalità di misurazione, il colore ecc. Questa risorsa presenta un riferimento al dispositivo mediante il quale vengono effettuate le misurazioni ed è referenziata dalle risorse Observation relative alla metrica stessa.

3.2.3 Observation

Le osservazioni sono un elemento centrale nell'assistenza sanitaria, utilizzate per supportare la diagnosi, monitorare i progressi, determinare pattern e persino catturare le caratteristiche demografiche. La maggior parte delle osservazioni sono semplici asserzioni di coppie nome-valore con alcuni metadati. Questa risorsa copre un insieme di casi d'uso molto eterogeneo che include i parametri vitali, ovvero le misurazioni registrate dal monitor in esame. Le osservazioni sono centrali nella modellazione FHIR in quanto possono presentare riferimenti alla metrica e al dispositivo così come al paziente ad esse associato.

Alla luce delle risorse appena descritte, in Figura 3.3 viene illustrata una prima modellazione del dispositivo e delle sue osservazioni con le risorse FHIR base.

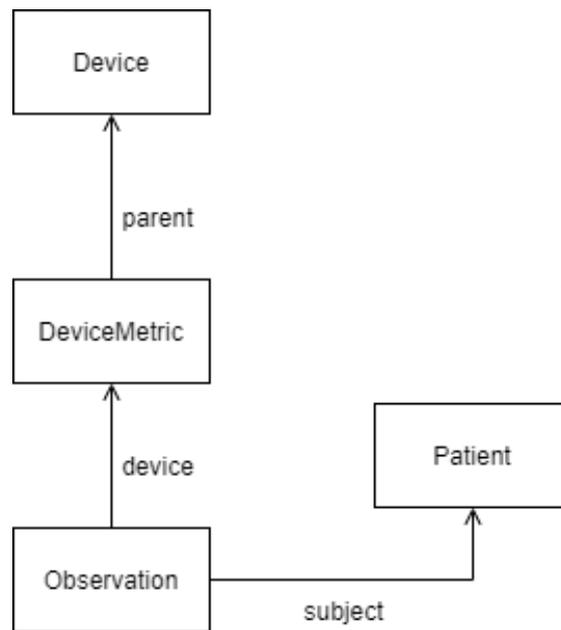


Figura 3.3: Modello semplificato con risorse FHIR

3.3 Secondo l'Implementation Guide

In questa sezione si vuole proporre una modellazione del dispositivo monitor parametri vitali Infinity M540 a partire dalla Point-of-Care Device Implementation Guide discussa nel secondo capitolo e sulla base del funzionamento del dispositivo precedentemente descritto.

Come abbiamo visto, l'Implementation Guide propone una modellazione del dispositivo a partire da un modello generico del Medical Device Communications Domain Information Model dello standard IEEE 11073. Questo modello prevede una rappresentazione gerarchica del dispositivo il quale, dal punto di vista logico, è un sistema globale con diversi sottosistemi distinti.

Come si può vedere dalla Figura 3.4 la struttura gerarchica relativa al dispositivo e ai dati che esso raccoglie si sviluppa a partire da cinque risorse logicamente collegate tra loro: MDS, VMD, Channel, Metric e Observation.

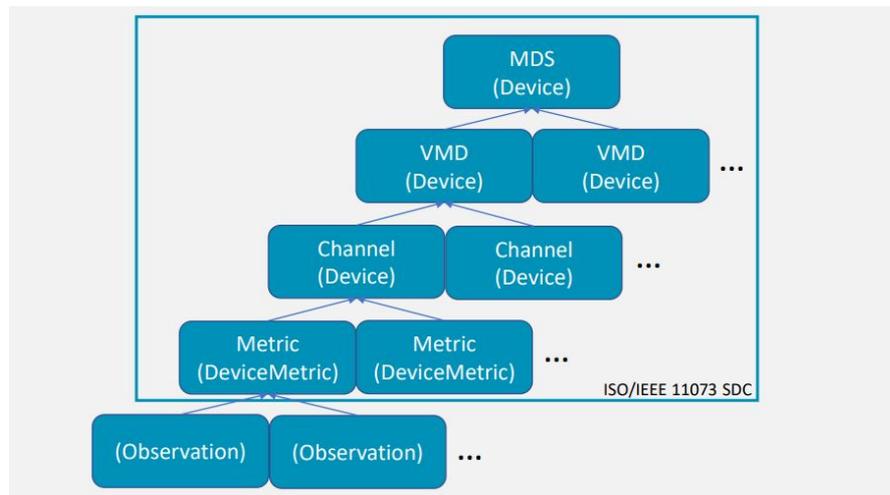


Figura 3.4: ISO/IEEE 11073 SDC modellato in HL7 FHIR [2]

3.3.1 Medical Device System

Il Medical Device System (MDS) è la risorsa di massimo livello nel modello gerarchico di un dispositivo Point-of-Care stabilendo il contesto generale per tutti i dati del dispositivo. Essendo un profilo della risorsa base Device, oltre a contenere tutte le informazioni relative al dispositivo stesso, definisce delle regole sulla nomenclatura da usare per i parametri "identifier" e "type" in accordo con le linee guida IEEE. Nel caso dell'M540 questa risorsa definisce il monitor parametri vitali nella sua globalità, ovvero circoscrive tutti quei sottosistemi incaricati di rilevare i parametri del paziente.

3.3.2 Virtual Medical Device

Un Virtual Medical Device (VMD) è una risorsa che rappresenta una particolare specializzazione del dispositivo, un sottosistema nel modello gerarchico di un PoCD. Le regole relative alla terminologia del campo type sono analoghe a quelle del MDS ed è inoltre specificato che la risorsa VMD non deve avere riferimenti al paziente o al luogo, al contrario, deve contenere un riferimento all'MDS di cui fa parte nel campo "parent".

Nel contesto del caso di studio, il monitor M540 è provvisto di più moduli per monitorare i diversi parametri vitali del paziente. In questo caso è opportuno

modellare ogni modulo con una risorsa VMD, in modo da suddividere il sistema in sottosistemi logicamente distinti. La suddivisione è stata effettuata similmente alla Figura 3.5:

- Blood pressure: per quanto riguarda la pressione sanguigna.
- Temperature: per gestire la temperatura corporea.
- Pulse-Oximeter: per la saturazione del sangue.
- ECG: per raggruppare tutti i dati relativi all'elettrocardiogramma.

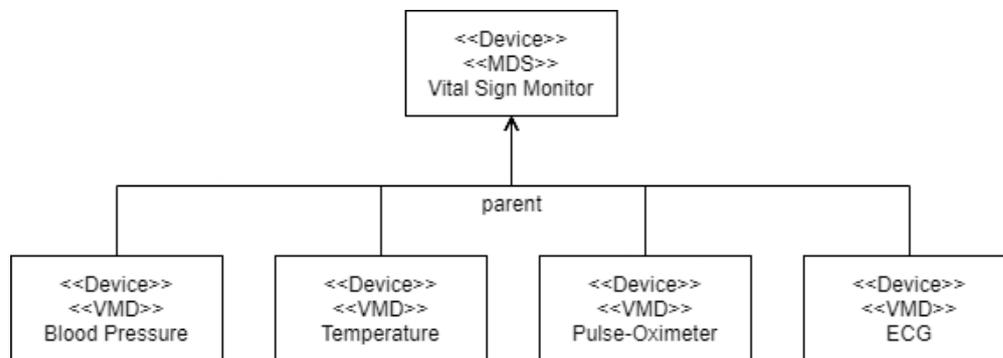


Figura 3.5: Diagramma UML delle risorse VMD

3.3.3 Channel Device

Si tratta di un profilo della risorsa base Device che rappresenta un canale e viene utilizzato per raggruppare insieme metriche per avere una suddivisione logica delle stesse. Le regole dettate da questa risorsa sono analoghe a quelle del profilo VMD se non per il fatto che un Channel Device deve avere un riferimento alla risorsa VMD padre. Inoltre, qualora più istanze dello stesso tipo dovessero essere presenti, l'Implementation Guide mette a disposizione l'estensione Instance per distinguerle.

Nel caso del monitor parametri vitali sono stati scelti i seguenti canali per raggruppare le metriche di ogni VMD.

- **Blood pressure**

Siccome tutte le misurazioni riguardanti la pressione sanguigna si possono suddividere logicamente in non invasive o invasive, sono stati creati due canali appositi relativi al VMD blood pressure, vedere Figura 3.6.

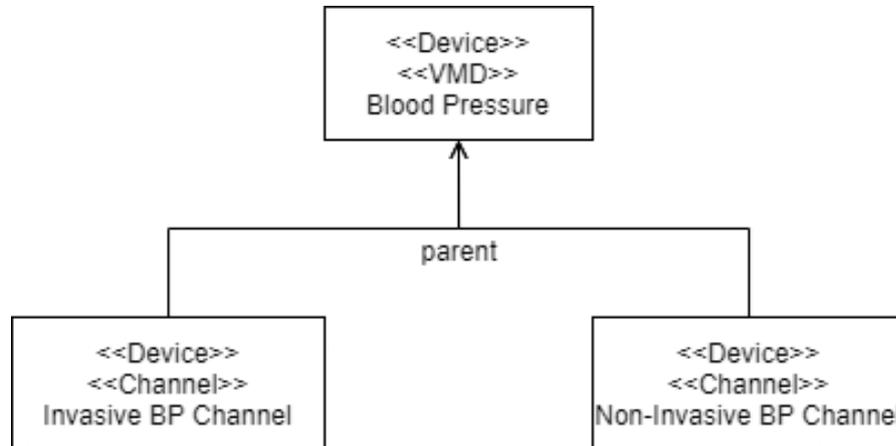


Figura 3.6: Diagramma UML canali pressione sanguigna

- **Temperature**

Per la temperatura corporea è stato scelto un solo canale, Figura 3.7, in quanto non vi è una distinzione logica delle relative metriche

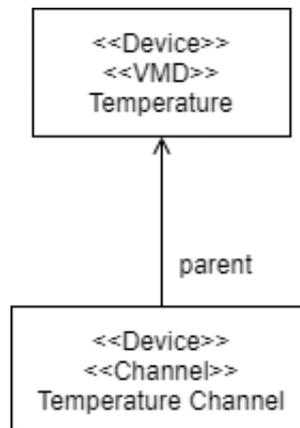


Figura 3.7: Diagramma UML canali temperatura corporea

- **Pulse-Oximeter**

I parametri vitali relativi alla saturazione sanguigna vengono raggruppati in un singolo canale come mostrato in Figura 3.8.

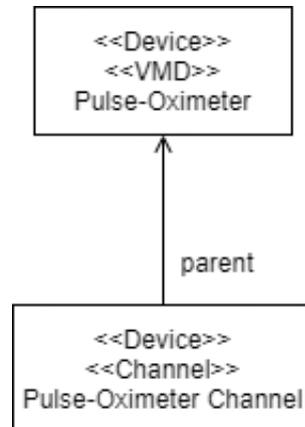


Figura 3.8: Diagramma UML canali pulsossimetro

- **ECG**

Le misurazioni relative all'ECG sono molteplici, per questo motivo è stata scelta una suddivisione in quattro canali: il primo raggruppa le informazioni riguardanti la frequenza respiratoria, il secondo la frequenza cardiaca, il terzo i parametri relativi all'aritmia e l'ultimo le restanti misurazioni Figura 3.9.

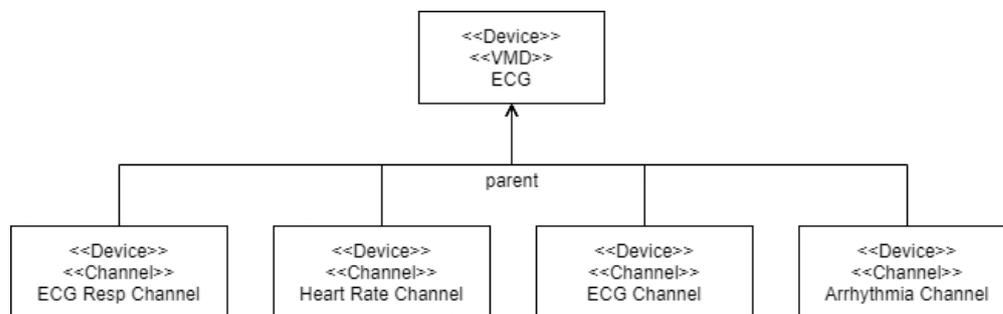


Figura 3.9: Diagramma UML canali ECG

3.3.4 Device Metric

L'Implementation Guide su cui si basa la modellazione mette a disposizione tre profili di questa risorsa base, tuttavia l'unico a cui si farà riferimento è il Numeric

DeviceMetric. Questo profilo rappresenta una misurazione numerica, un calcolo o le caratteristiche di un'impostazione. Da notare che l'unità di misura, se presente, deve fare riferimento al sistema di codifica ISO/IEEE 11073 MDC Units of measurement. Inoltre, il campo "source" deve puntare alla risorsa MDS del dispositivo mentre come padre la risorsa del canale a cui appartiene.

Di seguito la modellazione proposta per rappresentare le diverse metriche relative ai canali precedentemente illustrati.

- **Invasive BP Channel**

Fanno parte di questo canale tutte le metriche relative alla pressione sanguigna invasiva, anche detta pressione arteriosa. Essa si suddivide in sistolica, diastolica e media Figura 3.10.

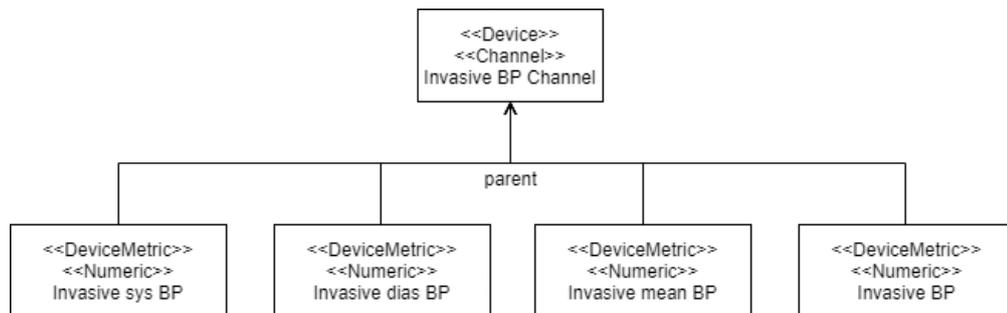


Figura 3.10: Diagramma UML metriche Invasive Blood Pressure

- **Non-Invasive BP Channel**

Fanno parte di questo canale tutte le metriche relative alla pressione sanguigna non invasiva. Anch'essa si suddivide in sistolica, diastolica e media Figura 3.11.

- **Temperature Channel**

In questo canale vengono raggruppate le metriche relative alla temperatura del paziente. In questo caso si è scelto di modellare tre diverse misurazioni: temperatura del corpo, temperatura della pelle e temperatura interna Figura 3.12.

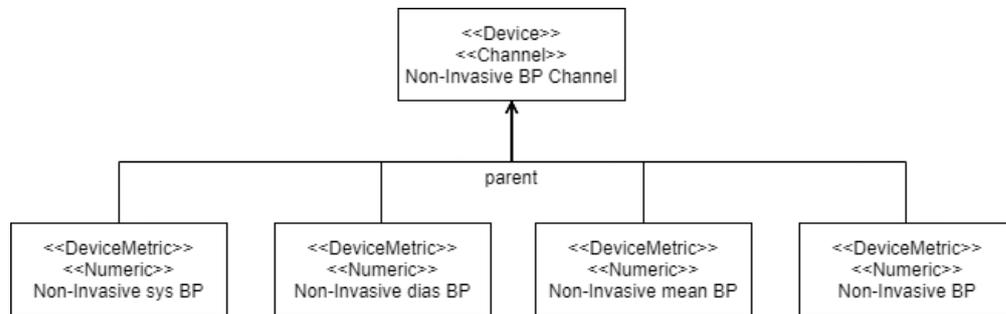


Figura 3.11: Diagramma UML metriche Non-Invasive Blood Pressure

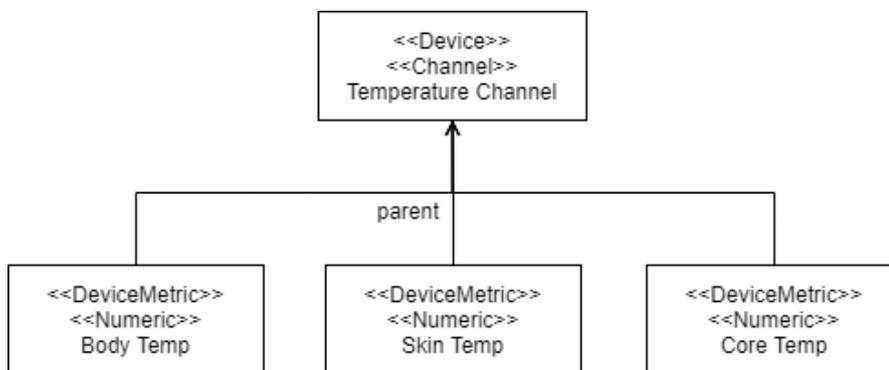


Figura 3.12: Diagramma UML metriche Temperature

- **Pulse-Ox Channel**

Questo canale convoglia le metriche del pulsossimetro, ovvero la saturazione sanguigna e l'intensità delle pulsazioni Figura 3.13.

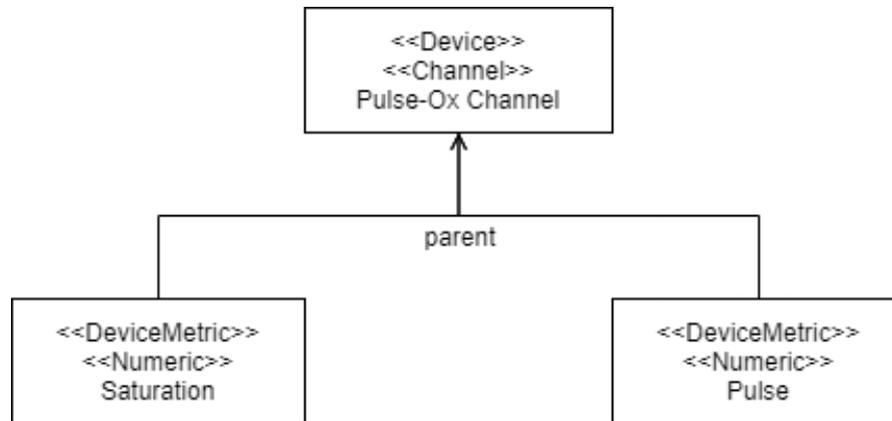


Figura 3.13: Diagramma UML metriche Pulse-Oximeter

- **ECG Resp Channel**

In questo canale vengono collezionate le metriche sulla frequenza respiratoria del paziente Figura 3.14.

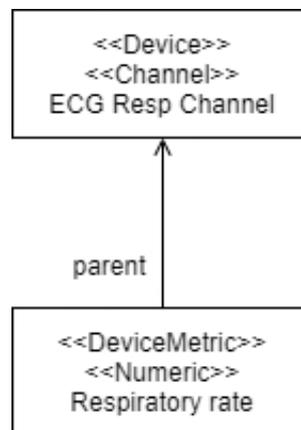


Figura 3.14: Diagramma UML metriche Respiratory Rate

- **Heart Rate Channel**

In questo canale vengono riunite le metriche relative alla frequenza cardiaca del paziente Figura 3.15.

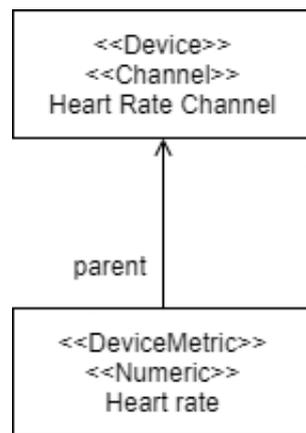


Figura 3.15: Diagramma UML metriche Heart Rate

- **Arrhythmia Channel**

In questo canale vengono raggruppate le metriche relative all'aritmia Figura 3.16.

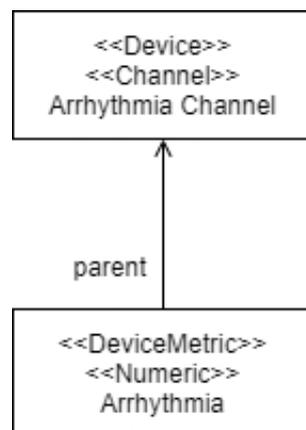


Figura 3.16: Diagramma UML metriche Arrhythmia

- **ECG Channel**

In questo canale vengono raggruppate le metriche relative alle rilevazioni dell'elettrocardiogramma 3.17.

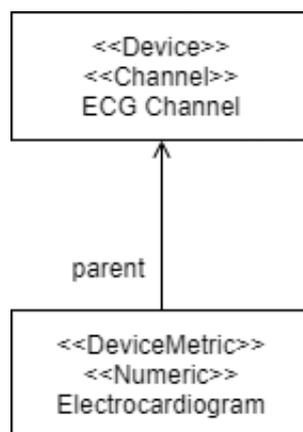


Figura 3.17: Diagramma UML metriche ECG

3.3.5 Observation

La PoCD Implementation Guide offre quattro diversi profili per questa risorsa base: Numeric Observation, Compound Numeric Observation, Enumeration Observation e Sample Array Observation. Siccome le metriche utilizzate in questa modellazione implementano il profilo Numeric DeviceMetric, è opportuno scegliere le relative Observation di tipo Numeric. Queste risorse richiedono l'utilizzo di codici conformi a ISO/IEEE 11073 MDC Metric nomenclature ed eventualmente a LOINC. Inoltre, il campo "subject" deve avere un riferimento alla risorsa MDS del dispositivo o alla risorsa relativa al paziente su cui è stata effettuata la misurazione, mentre il campo "device" deve riferirsi alla metrica associata.

Nella modellazione proposta ogni risorsa Observation è relativa ad una specifica metrica, ottenendo così il diagramma completo del sistema relativo al dispositivo monitor parametri vitali mostrato in Figura 3.18.

3.3.6 Conclusioni sul modello ottenuto

Al termine di questo capitolo si è ottenuta una possibile modellazione del dispositivo monitor parametri vitali e delle rispettive misurazioni mediante le risorse messe a disposizione da FHIR, in particolare dalla PoCD Implementation Guide. Nel corso del prossimo capitolo verrà analizzato un approccio interoperabile attraverso il quale sarà possibile comunicare sfruttando le risorse precedentemente descritte.

Verranno inoltre mostrate le specifiche tecnologie, come il FHIR Server, necessarie per il corretto funzionamento del sistema proposto.

Capitolo 4

Proposta di integrazione T4C

Dopo aver delineato una possibile modellazione del dispositivo monitor parametri vitali attraverso le risorse FHIR, in questo capitolo partendo dal progetto già esistente T4C, si vuole proporre una integrazione dello stesso che, in un contesto sanitario interoperabile, consenta alle applicazioni che si interfacciano al sistema di scambiare le informazioni mediante una rappresentazione conforme a FHIR. Si farà quindi un'analisi del modello offerto da T4C proseguendo con la presentazione di una possibile integrazione in un'ottica di interoperabilità.

4.1 Stato attuale: T4C e monitor parametri vitali

Il progetto di ricerca Tracking for Care (T4C) si cala in un'ottica di studio e applicazione di sistemi informatici innovativi a supporto delle attività del personale medico e infermieristico, in particolare facilitando il tracciamento e la documentazione durante emergenze tempo-dipendenti (ad esempio Trauma, Ictus, Infarto). Come brevemente accennato, al giorno d'oggi il progetto T4C fornisce già la possibilità alle applicazioni di interfacciarsi, mediante apposite API, alle informazioni relative ai dispositivi medici monitor parametri vitali ed emogasanalizzatore (EGA). Per far ciò, nel caso del monitor parametri vitali, è stato creato un web service HTTP implementato con l'ausilio di Vert.x che in maniera "solicited" reperisce le informazioni relative al monitor e alle sue misurazioni effettuando richieste

al gateway proprietario mediante appropriati messaggi QRY in formato HL7. In base alla risposta del gateway (sempre in formato HL7), viene poi opportunamente manipolato il database su cui si appoggia il servizio. Infine, il servizio stesso implementa specifiche API REST che permettono alle applicazioni sviluppate di interfacciarsi usufruendo delle seguenti funzionalità:

- `GET api/monitors/:monitor_id/vitalsigns`
ottenere i parametri vitali di uno specifico monitor a partire dal suo id;
- `GET api/monitors`
ottenere l'insieme dei monitor registrati;
- `POST api/monitors`
registrare un nuovo monitor
- `PUT, DELETE api/monitors/:monitor_id`
aggiornare o rimuovere le informazioni di uno specifico monitor;
- `POST api/sessions`
registrare una nuova sessione;
- `GET api/sessions/:session_id/start`
iniziare una sessione a partire dal suo id;
- `GET api/sessions/:session_id/stop`
interrompere una sessione a partire dal suo id;
- `GET, DELETE api/sessions/:session_id`
ottenere o eliminare le informazioni di una specifica sessione.

A partire dalla descrizione del servizio realizzato nel progetto T4C, si può ottenere una rappresentazione di un suo scenario nel diagramma illustrato in Figura 4.1. Nella prossima sezione si propone un'integrazione del modello presente verso una direzione FHIR oriented.

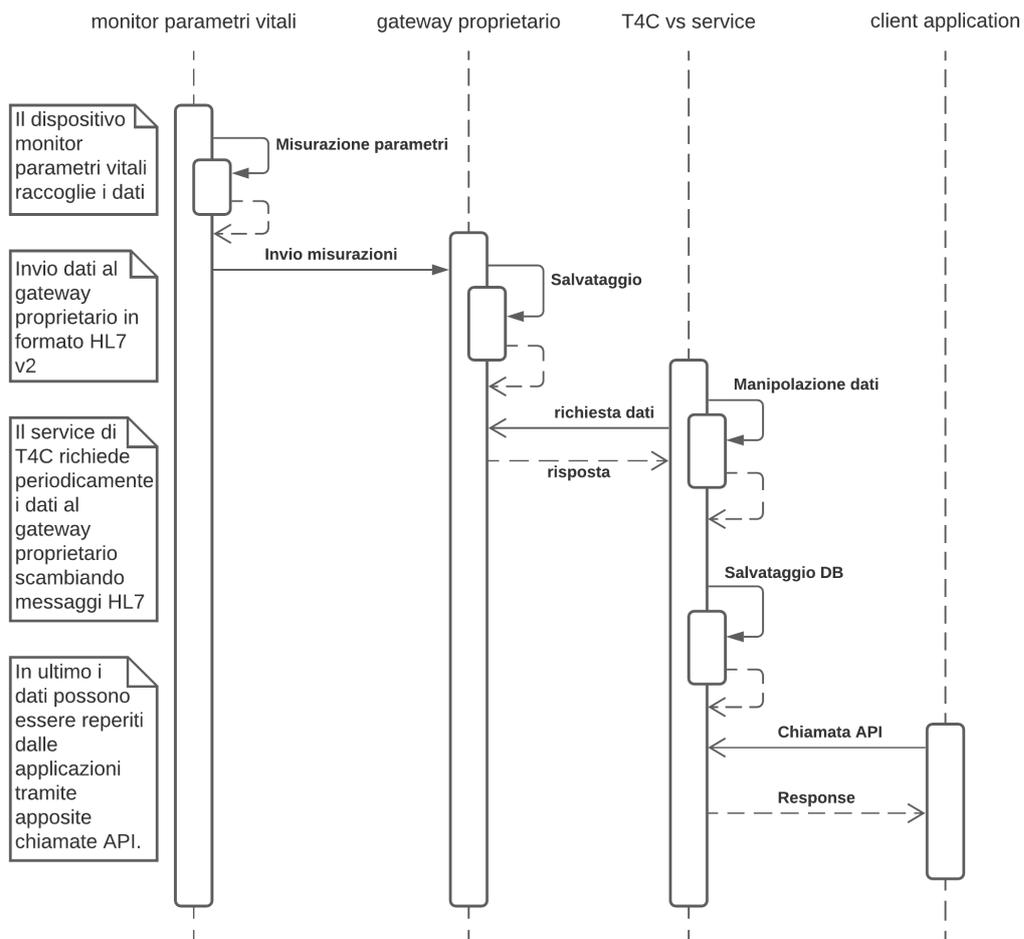


Figura 4.1: Diagramma di sequenza scambio dati con il servizio T4C

4.2 Proposta di integrazione verso un sistema FHIR interoperabile

In questa sezione viene avanzata una possibile proposta di integrazione al progetto T4C precedentemente descritto, con il fine di rendere interoperabile e conforme a FHIR l'interfacciamento al dispositivo preso in esame ovvero il monitor parametri vitali. L'obiettivo è quindi quello di creare un modulo che, integrato a T4C, permetta alle applicazioni che si interfacciano al sistema di reperire i dati relativi al monitor ed ai parametri vitali che esso registra come risorse FHIR andando quindi verso una direzione di interoperabilità. Questo componente si collocherebbe quindi tra il servizio di T4C e le applicazioni client, esponendo apposite API in grado di scambiare le informazioni secondo lo standard FHIR. Per far ciò è stato scelto di sfruttare un FHIR server, di seguito descritto.

4.2.1 FHIR Server

Un FHIR Server è in grado di elaborare, convalidare e archiviare dati sanitari in un formato standard del settore che può essere utilizzato per eseguire ricerche e altre funzionalità. Applicando la semantica e i modelli di dati standardizzati in conformità con la specifica FHIR, un FHIR Server semplifica la gestione delle informazioni presenti sui sistemi. Abilita funzionalità come l'analisi avanzata e l'apprendimento automatico, che possono beneficiare sulla salute e sull'impatto organizzativo. Un aspetto importante da tenere in considerazione è che il server deve essere alimentato di informazioni sotto forma di risorse FHIR affinché queste possano poi essere salvate e successivamente rese disponibili alle applicazioni che ne vogliono fare uso. Per quanto riguarda il salvataggio dei dati, solitamente un FHIR Server si appoggia a una struttura database. L'interfacciamento che offre un FHIR Server è effettuato mediante RESTful API e segue le direttive di FHIR, proponendo un'organizzazione dei dati per tipologia di risorsa. Ad esempio, effettuando una chiamata HTTP GET [base]/[type] verranno restituite tutte le risorse di tipo "type" presenti sul server stesso. Le API permettono di eseguire ricerche anche dettagliate rispetto alle risorse presenti sul server: il server determina quale insieme

di risorse che gestisce soddisfa i criteri specifici e restituisce i risultati nella risposta HTTP come un "bundle" che include le risorse risultato della ricerca.

4.2.2 Specifica dei requisiti

Di seguito verranno analizzati i requisiti che il componente proposto deve soddisfare.

Requisiti funzionali

Il software deve essere in grado di:

- gestire le informazioni ricevute dal servizio di T4C;
- modellare tali informazioni in modo conforme alle risorse FHIR;
- tradurre le risorse in formato JSON;
- connettersi ad un FHIR Server;
- registrare sul server le risorse create;
- permettere alle applicazioni di interfacciarsi attraverso apposite API. Queste API devono includere le funzionalità già messe a disposizione nel progetto T4C.

Requisiti non funzionali

Un aspetto fondamentale del software prodotto è quello dell'interoperabilità, ovvero deve consentire ad uno spettro eterogeneo di applicazioni di recuperare le informazioni relative al monitor parametri vitali. Inoltre l'applicativo deve essere efficiente, cioè deve ottimizzare l'utilizzo delle risorse ed affidabile, garantendo la correttezza dei dati gestiti.

4.2.3 Scenario sistema integrato

Alla luce di dei requisiti che deve soddisfare il componente proposto, si può dipingere uno scenario di funzionamento del sistema finale. Importante sottolineare come il modulo si collochi come un ponte tra il servizio già esistente di T4C e il livello applicativo, appoggiandosi ad un server per la gestione delle API Figura 4.2.

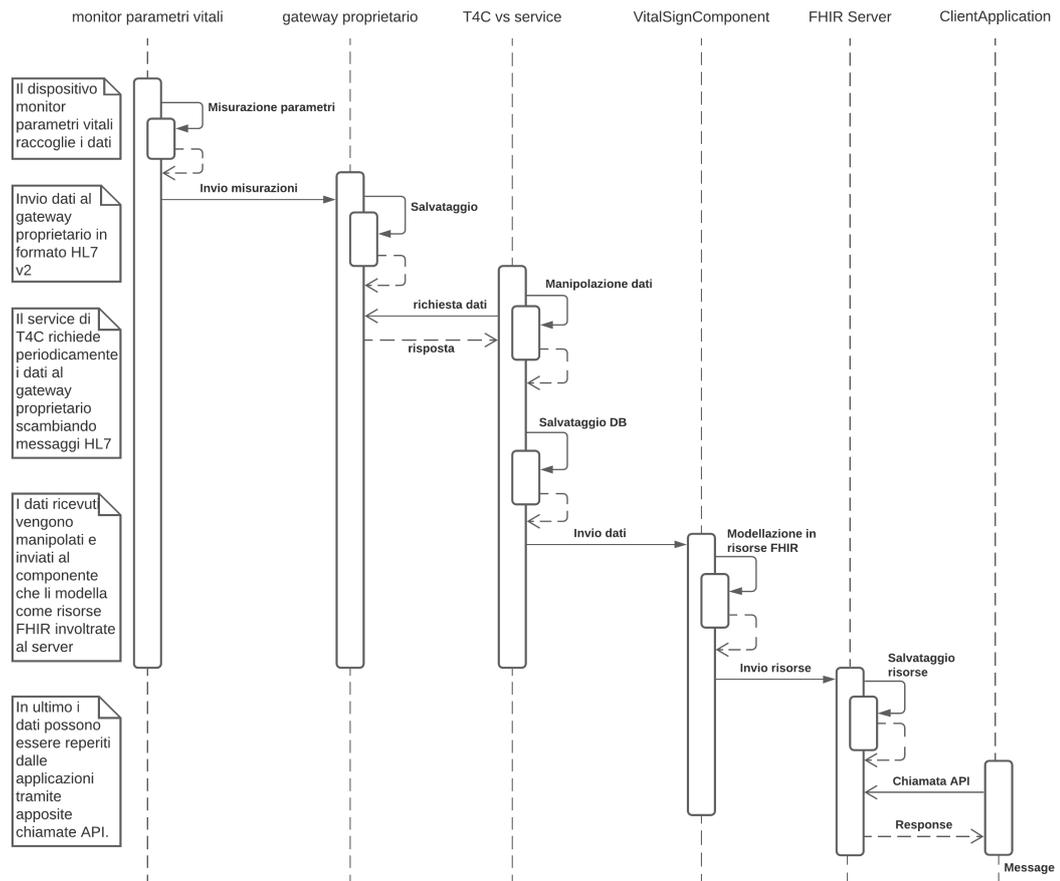


Figura 4.2: Diagramma di sequenza sistema integrato

4.3 Progettazione

Sulla base dell'analisi dei requisiti prodotta in fase di analisi, in questa sezione si vuole definire come essi saranno soddisfatti, delineando la struttura del software del sistema che verrà realizzato.

Come accennato precedentemente, il componente sviluppato deve integrarsi al già esistente servizio messo a disposizione nel progetto T4C. Quest'ultimo si interfaccia al gateway proprietario del monitor richiedendo periodicamente i segnali misurati e modellandoli in un contesto object oriented come mappa di VitalSign e VitalSignValue. VitalSign rappresenta i vari tipi di parametri vitali che possono essere registrati, incorporando concetti come il codice e il sistema di codifica associato (solitamente LOINC) mentre VitalSignValue delinea il valore e la sua unità di misura del parametro stesso. Ogni volta che il servizio riceve nuove misurazioni, queste vanno inoltrate al componente in esame attraverso il metodo `saveVitalSigns` che espone. Successivamente, il componente si appoggia al `FHIRParser`, elemento che si occupa di convertire le misurazioni in risorse FHIR come mostrato in Figura 4.3.

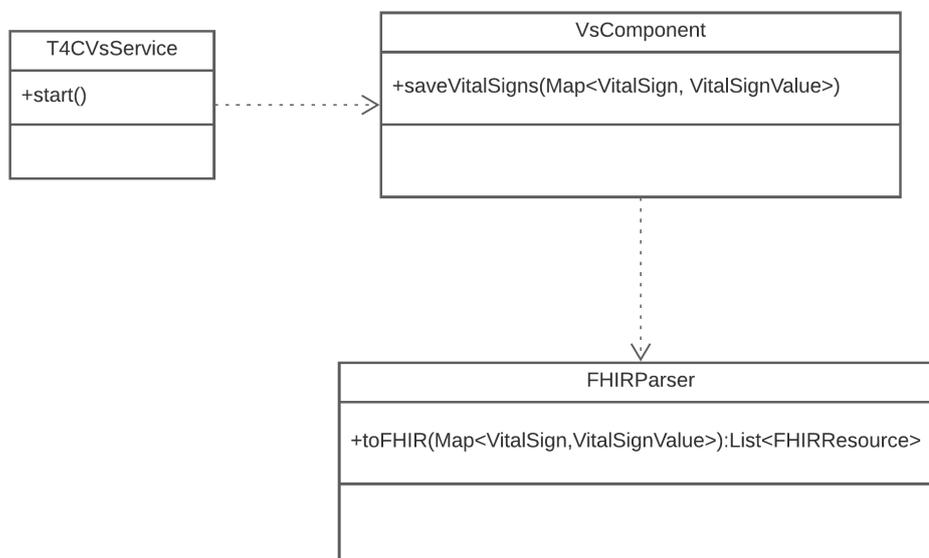


Figura 4.3: Schema UML componente integrato a T4C

Prima di procedere è bene illustrare come è stata modellata una risorsa FHIR. Una risorsa FHIR non è altro che un oggetto che, oltre a contenere tutti i campi della risorsa stessa, espone un metodo per ricavarne il contenuto informativo in formato JSON. Questa funzionalità è essenziale in quanto le informazioni scambiate tra client e server adottano questo formato. Nel contesto applicativo in esame sono state prese in considerazione le risorse Device e Observation, che fanno riferimento rispettivamente al monitor e alle misurazioni che esso rileva Figura 4.3.

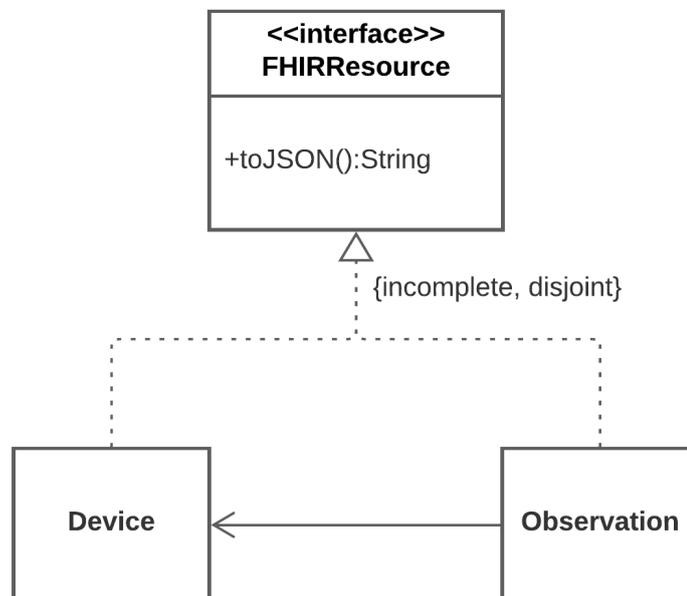


Figura 4.4: Schema UML modellazione risorse FHIR

Calando il sistema in un contesto client-server, si ottiene lo schema finale illustrato in Figura 4.5. Un aspetto importante è quello del Client, entità utilizzata dal VsComponent che permette di stabilire una connessione al server e di interfacciarsi ad esso registrando le risorse FHIR generate.

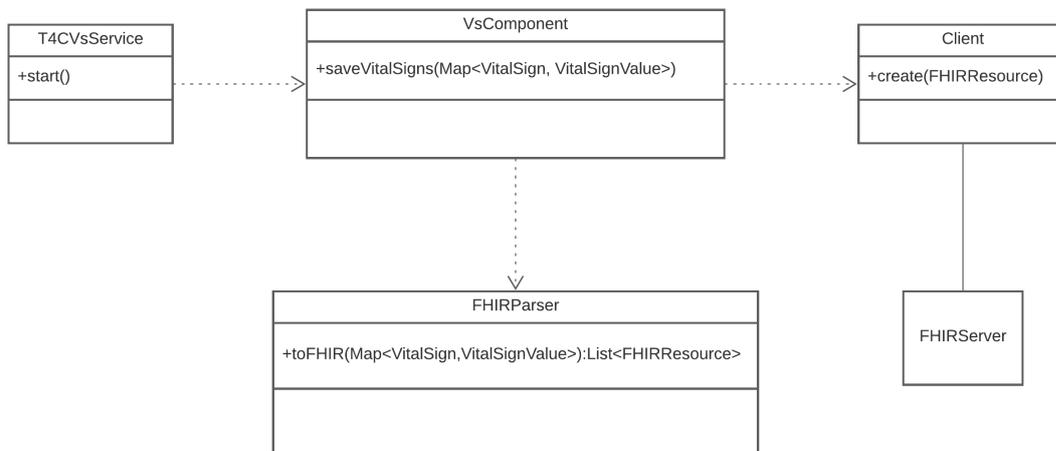


Figura 4.5: Diagramma UML finale del sistema

4.4 Sviluppo e testing

La fase di sviluppo attuata in questo progetto aderisce al modello a cascata, infatti l'analisi e la progettazione sono servite come input per la parte di implementazione.

In particolare, questa fase si è articolata come segue:

- Implementazione del componente di integrazione: nel corso di questa prima sottofase è stato implementato il modulo di integrazione a T4C, a partire dagli schemi realizzati durante la fase di progettazione.
- Validazione del software: attraverso test specifici è stato validato il lavoro svolto in fase di sviluppo.

4.4.1 Tecnologie utilizzate

Sviluppo componente

Per quanto riguarda lo sviluppo del componente, in linea col servizio realizzato in T4C, si è scelto di utilizzare il linguaggio **Java**. Java è un linguaggio di programmazione general-purpose, class-based e object-oriented. È tra i linguaggi di programmazione più utilizzati ed è sfruttato come piattaforma informatica per lo sviluppo di applicazioni.

Per la modellazione delle risorse FHIR e l'interfacciamento al server si è scelto di utilizzare la libreria Java **HAPI FHIR**. HAPI è stato progettato con l'intento principale di fornire un modo flessibile per aggiungere funzionalità FHIR alle applicazioni e fornisce un'implementazione completa dello standard HL7 FHIR per l'interoperabilità sanitaria in Java.

Ogni tipo di risorsa definita da FHIR ha una classe corrispondente, che contiene un numero di getter e setter per le proprietà di base di quella risorsa. HAPI cerca di rendere più facile il popolamento degli oggetti, fornendo molti metodi convenienti. Esistono metodi per utilizzare il tipo di dati FHIR effettivo, ma anche metodi pratici che utilizzano tipi built-in di Java.

HAPI FHIR ha il supporto integrato per i formati di codifica FHIR JSON e XML. Un parser integrato può essere utilizzato per convertire gli oggetti Java HAPI FHIR in una forma serializzata e per convertire i dati serializzati in oggetti Java. Infine, fornisce un meccanismo integrato per la connessione ai server FHIR REST.

Testing

Innanzitutto per verificare il corretto funzionamento del software è stato necessario utilizzare un FHIR Server. In particolare si è scelto di installare in locale l'immagine del **FHIR Server di IBM**, avvelendosi di un **docker**. IBM FHIR Server non è altro che un'implementazione Java modulare della versione 4 della specifica di HL7 FHIR con particolare attenzione alle prestazioni e alla configurabilità. La tecnologia Docker sfrutta il kernel di Linux e le sue funzionalità per isolare i processi in modo da poterli eseguire in maniera indipendente. Questa indipendenza è l'obiettivo dei container: la capacità di eseguire più processi e applicazioni in modo separato per sfruttare al meglio l'infrastruttura esistente pur conservando il livello di sicurezza che sarebbe garantito dalla presenza di sistemi separati.

Gli strumenti per la creazione di container, come Docker, consentono lo sviluppo a partire da un'immagine semplificando la condivisione di un'applicazione o di un insieme di servizi, con tutte le loro dipendenze, nei vari ambienti. Docker automatizza anche la distribuzione dell'applicazione (o dei processi che compongono un'applicazione) all'interno dell'ambiente containerizzato [4].

Infine, per verificare che il server venisse gestito correttamente, sono state effet-

tuare chiamate API con l'ausilio di **Postman**. Postman è uno strumento utilizzato per i test API. Essenzialmente consiste in un client HTTP che testa le richieste HTTP, utilizzando un'interfaccia utente, attraverso la quale vengono ottenuti diversi tipi di risposte che devono essere successivamente convalidate.

4.4.2 Testing

In questa fase è stato testato il corretto funzionamento del componente sviluppato, nonché le funzionalità delineate nell'analisi dei requisiti.

In primo luogo, date le complicazioni derivate dall'uso del servizio di T4C si è scelto di simulare delle possibili misurazioni registrate dal monitor. Per fare ciò è stata creata una classe `VitalSignGenerator` che implementa un metodo per creare dei valori di input realistici casuali, che vengono poi passati al componente vero e proprio. Quest'ultimo inizialmente attraverso il `FHIRParser` converte i parametri vitali in risorse FHIR, passate poi al client che permette la registrazione sul FHIR Server. Come FHIR Server è stata installata in locale l'immagine dell'IBM FHIR Server mediante un docker. Infine, con l'ausilio di Postman, è stato possibile verificare sia il corretto caricamento sul server delle informazioni registrate sotto forma di risorse FHIR, che il corretto funzionamento delle API messe a disposizione dal server stesso. Di seguito un esempio di come le misurazioni effettuate dal monitor vengono rappresentate in risorse FHIR:

Listato 4.1: Esempio osservazione in formato JSON caricata sul server

```
1 {  
2   "resourceType": "Observation",  
3   "status": "preliminary",  
4   "code": {  
5     "coding": [  
6       {  
7         "system": "http://loinc.org",  
8         "code": "8496-2"  
9       }  
10    ]  
11  },
```

```
12  "subject": {
13    "reference": "Device/17bda269229-e18a36ae-bd36
14    -4957-95c2-2f71d50cf397"
15  },
16  "valueQuantity": {
17    "value": 70,
18    "unit": "mmHg"
19  }
}
```

Questa stringa in formato JSON modella come una delle possibili osservazioni che possono essere rilevate dal monitor a parametri vitali viene salvata all'interno del FHIR Server. In questo esempio si tratta di una misurazione relativa alla pressione diastolica.

Tra le varie API messe a disposizione dal FHIR Server di seguito vengono riportate quelle più rilevanti e significative per il caso di studio:

- `POST [base]/Device`
per registrare un nuovo dispositivo;
- `GET [base]/Device`
per ottenere tutti i dispositivi presenti;
- `GET [base]/Device/monitor_id`
per ottenere uno specifico monitor a partire dal suo id;
- `PUT [base]/Device/monitor_id`
per modificare le informazioni di un monitor;
- `DELETE [base]/Device/monitor_id`
per eliminare uno specifico monitor a partire dal suo id;
- `POST [base]/Observation`
per registrare una nuova misurazione;

- GET [base]/Observation
per ottenere tutte le misurazioni di tutti i dispositivi;
- GET [base]/Observation?subject=Device/monitor_id;
per ottenere tutte le misurazioni effettuate da uno specifico monitor a partire dal suo id.

Ad esempio, facendo una chiamata API per ottenere tutti i dispositivi presenti, il server risponderà restituendo una risorsa di tipo "Bundle" che funge da contenitore di altre risorse, contenente l'insieme dei monitor, sempre rappresentati come risorse FHIR in formato JSON, vedere Listato 4.2.

Listato 4.2: Esempio risposta del server alla chiamata API per ottenere tutti i monitor

```
1 {
2   "resourceType": "Bundle",
3   "id": "e768e914-1840-4753-b615-7fb2fe4650b7",
4   "type": "searchset",
5   "total": n,
6   "link": [
7     {
8       "relation": "self",
9       "url": "https://localhost:9443/fhir-server/
10         api/v4/Device?_count=10&_page=1"
11     }
12 ],
13   "entry": [
14     Device1,
15     Device2,
16     ...
17 ]
18 }
```

Per una migliore formattazione la lista dei monitor è stata rappresentata in modo semplificato.

4.4.3 Ottica ad eventi

È interessante considerare come ci siano due principali approcci per lo sviluppo di applicazioni che si basano sul sistema proposto: il primo, più semplice, prevede che le applicazioni facendo "polling", ovvero richieste periodiche, interroghino il FHIR Server con le apposite API verificando poi eventuali cambiamenti. La seconda modalità, quella ad eventi, attraverso un meccanismo "publish/subscribe" consente alle applicazioni di registrarsi ad un evento (ad esempio la creazione, modifica o cancellazione di una risorsa) ricevendo una notifica ogni qualvolta l'evento in questione si verifica. Quest'ultimo approccio, anche chiamato event-driven, offre un'ottimizzazione maggiore delle risorse e garantisce un tracciamento continuo delle informazioni ed è applicabile in un FHIR Server. Infatti, l'IBM FHIR Server fornisce un servizio di notifica sulle operazioni di creazione, aggiornamento ed eliminazione. Esso può essere utilizzato da altri componenti sanitari per scatenare azioni specifiche che devono essere eseguite durante l'aggiornamento delle risorse gestite dal server FHIR. Il servizio di notifica supporta due implementazioni: WebSocket e Kafka.

Conclusioni e sviluppi futuri

In questa tesi è stato affrontato il problema dell'interoperabilità tra dispositivi medici in un contesto IoMT, concentrandosi sul monitor parametri vitali. In particolare, dopo aver proposto una modellazione del dispositivo secondo lo standard FHIR, ci si è calati nel contesto applicativo del già esistente progetto T4C, proponendo un software che, integrandosi al servizio relativo al monitor, proponesse una gestione e un interfacciamento dei dati sanitari prodotti dal monitor stesso in un formato standard, rendendo la soluzione il più interoperabile possibile. Il componente proposto fornisce quindi degli strumenti per gestire i monitor in uso e i parametri vitali da essi misurati, formattandoli secondo lo standard FHIR.

Nonostante la soluzione proposta offra dei miglioramenti in ottica di interoperabilità, ci sono alcuni aspetti da tenere in considerazione. Innanzitutto, le attuali applicazioni sviluppate che si interfacciano alle API messe a disposizione dal servizio di T4C, dovranno adeguarsi alle API FHIR se vogliono sfruttarne i benefici.

Altro aspetto fondamentale da considerare è quello del contesto in cui installare il FHIR Server. Infatti, se da un lato installare un FHIR Server per ogni reparto dell'ospedale può non essere la scelta ottimale, avere un unico FHIR Server a livello ospedaliero comporterebbe complicazioni per quanto riguarda la suddivisione logica delle informazioni. Bisognerà quindi valutare, a partire da una visione dell'intero sistema, come organizzare dal punto di vista digitale l'insieme dei servizi aderendo ad un approccio uniforme.

Infine, il sistema proposto è stato sviluppato come componente direttamente integrato al servizio di T4C. Questa soluzione garantisce una coesione dell'insieme maggiore e performance superiori. Tuttavia, un altro metodo che si potrebbe

scegliere è quello di creare un ulteriore servizio indipendente seguendo un approccio più disaccoppiato.

Ringraziamenti

Tengo a ringraziare innanzitutto il professore Alessandro Ricci per l'occasione offertami di contribuire ad un progetto attuale ed innovativo, e il professore Angelo Croatti per la continua disponibilità e assistenza fornita durante l'intero percorso. Infine, ringrazio tutte le persone che mi hanno supportato, in particolare il collega Edoardo Maioli con cui ho svolto il lavoro integrato.

Bibliografia

- [1] Fhir implementation guide. <https://www.hl7.org/fhir/implementationguide.html>. Ultimo accesso: 2021-08-26.
- [2] Hl7 fhir devdays. https://www.devdays.com/wp-content/uploads/2020/12/azpkt_201120_KathrinPiaRiechandTobiasWildi_Lets_build_Querying_Medical_Device_Data_For_Clinical_and_Workflow_Applications.pdf. Ultimo accesso: 2021-08-29.
- [3] Point-of-care device implementation guide. <https://build.fhir.org/ig/HL7/uv-pocd/index.html>. Ultimo accesso: 2021-08-26.
- [4] What is docker. <https://www.redhat.com/it/topics/containers/what-is-docker>. Ultimo accesso: 2021-09-16.
- [5] 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.
- [6] Faisal Alsubaei, Abdullah Abuhussein, and Sajjan Shiva. Security and privacy in the internet of medical things: Taxonomy and risk assessment. In *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, pages 112–120, 2017.
- [7] Riham Altawy and Amr M. Youssef. Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices. *IEEE Access*, 4:959–979, 2016.

- [8] A. Bagula, C. Lubamba, M. Mandava, H. Bagula, M. Zennaro, and E. Pietrosemoli. Cloud based patient prioritization as service in public health care. In *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, pages 1–8, 2016.
- [9] B. Bergh, A. Brandner, J. Heiß, U. Kutscha, A. Merzweiler, R. Pahontu, B. Schreiweis, N. Yüksesogul, T. Bronsch, and O. Heinze. Die rolle von integrating the healthcare enterprise (IHE) in der telemedizin. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz*, 58(10):1086–1093, September 2015.
- [10] Hemantha Krishna Bharadwaj, Aayush Agarwal, Vinay Chamola, Naga Rajiv Lakkaniga, Vikas Hassija, Mohsen Guizani, and Biplab Sikdar. A review on the role of machine learning in enabling iot based healthcare applications. *IEEE Access*, 9:38859–38890, 2021.
- [11] G. Calcagnini, E. Mattei, F. Censi, M. Triventi, R. Lo Sterzo, E. Marchetta, and P. Bartolini. Electromagnetic compatibility of wifi technology with life-supporting medical devices. In Olaf Dössel and Wolfgang C. Schlegel, editors, *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, pages 616–619, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [12] Claudia-Melania Chituc. An analysis of iot interoperability standards in the healthcare sector. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 2910–2915, 2019.
- [13] Nancy Connor, Deanne McArthur, and Pilar Camargo Plazas. Reflections on vital sign measurement in nursing practice. *Nursing Philosophy*, 22(1):e12326, 2021.
- [14] L. Minh Dang, Md. Jalil Piran, Dongil Han, Kyungbok Min, and Hyeonjoon Moon. A survey on internet of things and cloud computing for healthcare. *Electronics*, 8(7), 2019.

- [15] Hossein Fotouhi, Aida Causevic, Kristina Lundqvist, and Mats Björkman. Communication and security in health monitoring systems – a review. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 545–554, 2016.
- [16] Eric Freudenthal, David Herrera, Frederick Kautz, Carlos Natividad, Alexandria Ogrey, Justin Sipla, Abimael Sosa, Carlos Betancourt, and Leonardo Estevez. Suitability of nfc for medical device communication and power delivery. In *2007 IEEE Dallas Engineering in Medicine and Biology Workshop*, pages 51–54, 2007.
- [17] Sugondo Hadiyoso, Rohmat Tulloh, Yuyun Siti Rohmah, Akhmad Alfaruq, et al. Design and development armband vital sign monitor for health-care monitoring. *Journal of Medical Signals & Sensors*, 11(3):208, 2021.
- [18] Longfei Han, Senlin Luo, Jianmin Yu, Limin Pan, and Songjing Chen. Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes. *IEEE journal of biomedical and health informatics*, 19(2):728–734, 2014.
- [19] Alejandro Lopez-Rincon, Alberto Tonda, Lucero Mendoza-Maldonado, Eric Claassen, Johan Garssen, and Aletta D Kraneveld. Accurate identification of sars-cov-2 from viral genome sequences using deep learning. *bioRxiv*, 2020.
- [20] Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel, and Bernd Freisleben. Iot-based health monitoring via lorawan. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pages 519–524. IEEE, 2017.
- [21] Roberto Minerva, Abyi Biru, and Domenico Rotondi. Towards a definition of the internet of things (iot). *IEEE Internet Initiative*, 1(1):1–86, 2015.
- [22] A Rozlosnik, SI Termografía Infrarroja, and Sánchez de Bustamante. Potential contribution of the infrared industry in the future of iot/iiot. In *Proceedings of the 14th Quantitative InfraRed Thermography Conference, Berlin, Germany*, pages 25–29, 2018.

- [23] Abhishek Singh, Ashish Payal, and Sourabh Bharti. A walkthrough of the emerging iot paradigm: Visualizing inside functionalities, key features, and open issues. *Journal of Network and Computer Applications*, 143:111–151, 2019.
- [24] Chia-Rong Su, Jeyhun Hajiyev, Changjui James Fu, Kuo-Chin Kao, Chih-Hao Chang, and Ching-Ter Chang. A novel framework for a remote patient monitoring (rpm) system with abnormality detection. *Health Policy and Technology*, 8(2):157–170, 2019.
- [25] Abdulhamit Subasi, Emina Alickovic, and Jasmin Kevric. Diagnosis of chronic kidney disease by using random forest. In *CMBEBIH 2017*, pages 589–594. Springer, 2017.
- [26] Dinesh Valluru and I Jeya. Iot with cloud based lung cancer diagnosis model using optimal support vector machine. *Health care management science*, 23(4):670–679, 2020.
- [27] Diana C. Yacchirema, Carlos E. Palau, and Manuel Esteve. Enable iot interoperability in ambient assisted living: Active and healthy aging scenarios. In *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 53–58, 2017.
- [28] Jaeseok Yun, Il-Yeup Ahn, JaeSeung Song, and Jaeho Kim. Implementation of sensing and actuation capabilities for iot devices using onem2m platforms. *Sensors*, 19(20), 2019.
- [29] Tang Zhong, Cheng Mengjin, Zeng Peng, and Wang Hong. Real-time communication in wia-pa industrial wireless networks. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 2, pages 600–605. IEEE, 2010.