

RosettaEdge: Un'Architettura Mobile Edge Zero-Cloud per l'Interoperabilità Semantica tra Fascicolo Sanitario Elettronico (HL7 CDA) e Apple HealthKit

Specifica Ingegneristica, Modello di Minaccia, Transcodifica
Ontologica e Supporto Decisionale On-Device

Alessandro Mainardi 

Dipartimento di Informatica — Scienza e Ingegneria (DISI)
Alma Mater Studiorum — Università di Bologna, Bologna, Italia
ORCID: [0009-0003-7003-7247](https://orcid.org/0009-0003-7003-7247) · Wikidata: [Q141886579](https://wikidata.org/Q141886579)
Concept DOI: [10.5281/zenodo.23781231](https://doi.org/10.5281/zenodo.23781231) · GitHub: [@AlexMaina05](https://github.com/AlexMaina05)
Contatto: maina@alexmaina.dev · Portale: <https://alexmaina.dev/>

Ottobre 2026

Sommario

La digitalizzazione della sanità pubblica ha introdotto infrastrutture complesse per la raccolta e conservazione della documentazione medica, tra cui il Fascicolo Sanitario Elettronico (FSE 2.0) in Italia. Ciononostante, persiste un divario tra la produzione istituzionale dei referti clinici e la loro effettiva fruibilità da parte del cittadino-paziente. La maggior parte dei dati resta segregata in documenti statici (PDF) o strutturata secondo lo standard HL7 Clinical Document Architecture (CDA Release 2), il cui contenuto semantico risulta inaccessibile agli ecosistemi di monitoraggio fisiologico consumer. Questo lavoro presenta **RosettaEdge**, un'architettura software nativa per piattaforme Apple (iOS e watchOS) concepita per superare tale frammentazione attraverso un modello di elaborazione locale (*Zero-Cloud*) conforme ai principi di *Privacy-by-Design*. Il sistema realizza una pipeline deterministica interamente locale in grado di elaborare documenti HL7 CDA R2, estrarne le osservazioni cliniche codificate in formato LOINC ed eseguirne la normalizzazione dimensionale (UCUM) e la transcodifica diretta verso il framework Apple HealthKit per i parametri fisiologici supportati. Per prevenire la contaminazione dei dati personali tra familiari (frequente nella gestione da parte di caregiver), il motore integra un *Identity Firewall* basato sulla validazione algebrica del Codice Fiscale e su euristiche di string matching (distanza di Jaro-Winkler). Quale meccanismo di fallback per referti non strutturati o cartacei, viene implementata una pipeline di Vision OCR guidata dal paradigma *Human-in-the-Loop*. La persistenza a riposo adotta la Data Protection API di iOS (Classe A) combinata con chiavi biometriche custodite nel Secure Enclave (Face ID/Touch ID) per il controllo degli accessi all'archivio locale. Infine, un motore di *Clinical Decision Support* (CDS) locale esegue analisi temporali e trend longitudinali (pressione arteriosa, profilo glicemico) secondo le linee guida internazionali (ESC/ESH e ADA) operando esclusivamente on-device senza dipendenze da server terzi. Il lavoro formalizza l'architettura ingegneristica, ne analizza il profilo di conformità al GDPR (Art. 25 e 32) e valida sperimentalmente l'accuratezza semantica e le prestazioni computazionali attraverso una suite di test unitari, avversari e benchmark empirici su terminali fisici.

Keywords: Fascicolo Sanitario Elettronico (FSE 2.0), HL7 CDA Release 2, LOINC, Apple HealthKit, Zero-Cloud Architecture, Privacy-by-Design, GDPR Art. 9, Mobile Health (mHealth), Clinical Decision Support (CDS), Swift Concurrency.

Abstract

The digitization of public healthcare has introduced comprehensive architectures for medical records, such as the Italian Electronic Health Record (FSE 2.0). However, a significant gap remains between institutional clinical data and actionable patient self-monitoring. Most clinical observations remain locked inside static PDF files or structured within HL7 Clinical Document Architecture (CDA Release 2) envelopes whose semantic payload is inaccessible to modern consumer health ecosystems. This paper introduces **RosettaEdge**, a native zero-cloud mobile edge software architecture for Apple platforms (iOS and watchOS) designed to bridge this divide following Privacy-by-Design and Privacy-by-Default principles. The system executes a deterministic on-device pipeline capable of ingesting HL7 CDA R2 XML payloads, extracting LOINC-coded laboratory observations, resolving dimension units via UCUM, and mapping them directly to Apple HealthKit data types for supported clinical analytes. To eliminate cross-contamination across shared family medical records, the engine deploys an on-device Identity Firewall leveraging formal fiscal identifier verification and Jaro-Winkler string distance heuristics. For unstructured or scanned paper documents, a Vision OCR fallback pipeline with Human-in-the-Loop verification is integrated. At-rest data persistence leverages iOS Data Protection (Class A) combined with Secure Enclave hardware-gated biometric access control (Face ID/Touch ID). Furthermore, an on-device Clinical Decision Support (CDS) engine evaluates longitudinal physiological time series (blood pressure, blood glucose) against established international guidelines (ESC/ESH and ADA) without external cloud dependencies. This paper details the formal system design, provides a thorough GDPR compliance analysis (Articles 25 and 32), and validates performance and semantic accuracy through automated test suites and on-device hardware benchmarks.

Keywords: Electronic Health Records, HL7 CDA R2, LOINC, Apple HealthKit, Zero-Cloud Computing, Privacy-by-Design, Mobile Health, Clinical Decision Support, Edge Computing.

Scheda Bibliografica & Identificatori Digitali Persistenti (Metadata Attribution)

Autore Principale:	Alessandro Mainardi  (maina@alexmaina.dev)
Affiliazione Istituzionale:	Alma Mater Studiorum — Università di Bologna, Dipartimento di Informatica — Scienza e Ingegneria (DISI)
Identificatore ORCID:	https://orcid.org/0009-0003-7003-7247
Entità Wikidata:	Q141886579 (<i>Alessandro Mainardi</i>)
Concept Digital Object Identifier (DOI):	10.5281/zenodo.23781231 (Archivio Accademico Zenodo / CERN / OpenAIRE)
Codice Sorgente & Repository:	https://github.com/AlexMaina05 · Portale Personale: https://alexmaina.dev/
Classificazione ACM CCS (2012):	Applied computing → Life and medical sciences → Health informatics; Security and privacy → Privacy-preserving protocols; Software and its engineering → Software architectures.
Dichiarazione sul Copyright:	© 2026 Alessandro Mainardi. Quest'opera è distribuita sotto i termini della Licenza Internazionale Creative Commons Attribuzione 4.0 (CC BY 4.0).

Indice

1	Introduzione e Inquadramento del Problema	5
1.1	Il Paradosso della Sanità Digitale e i Silos Informativi	5
1.2	La Sovranità del Dato e l'Imperativo Zero-Cloud	5
1.3	Obiettivi e Contributi del Progetto RosettaEdge	6
1.4	Struttura della Monografia	7
2	Stato dell'Arte e Fondamenti Teorici	7
2.1	L'Evoluzione degli Standard di Interoperabilità Sanitaria	7
2.2	Anatomia del Documento HL7 CDA Release 2	8
2.3	L'Infrastruttura Nazionale di Interoperabilità (INI) e i Sistemi Regionali	9
2.4	Ontologie e Sistemi di Codifica Clinica	10
2.4.1	LOINC (Logical Observation Identifiers Names and Codes)	10
2.4.2	UCUM (Unified Code for Units of Measure)	10
2.4.3	ICD (International Classification of Diseases)	10
2.5	Ecosistemi di Mobile Health: Apple HealthKit vs Google Health Connect	10
2.6	Analisi Comparativa Multidimensionale degli Standard	10
3	Architettura di Sistema e Modello di Sicurezza Zero-Cloud	11
3.1	Il Paradigma Zero-Cloud e l'Edge Computing Sanitario	11
3.2	Analisi di Conformità al Regolamento Generale sulla Protezione dei Dati (GDPR)	12
3.2.1	Valutazione d'Impatto sulla Protezione dei Dati (DPIA)	13
3.3	Modellazione delle Minacce (STRIDE Threat Model)	13
3.4	Architettura di Sicurezza Hardware: Apple Secure Enclave e Gerarchia delle Chiavi	14
3.4.1	Apple Keychain Services e Secure Enclave	14
3.4.2	SwiftData e Data Protection API	14
4	Il Motore di Transcodifica Semantica (Rosetta Engine)	15
4.1	Architettura del Parser XML: Streaming SAX vs Document Object Model	15
4.2	Implementazione dello Streaming Parser SAX in Swift	15
4.3	Formalizzazione del Mapping Ontologico LOINC → HealthKit	17
4.4	Architettura a Doppio Binario Semantico: Telemetria vs. Chimica Clinica	18
4.5	Normalizzazione Dimensionale e Risoluzione delle Unità UCUM	18
4.6	L'Algoritmo di Identity Firewall	19
4.6.1	Validazione Aritmetica del Codice Fiscale Italiano	19
4.6.2	Fuzzy Name Matching con Metrica di Jaro-Winkler	20
4.7	Sottosistema Vision OCR e Modello Human-in-the-Loop	20
5	Algoritmi di Supporto Decisionale Clinico (CDS) On-Device	21
5.1	Obiettivi e Limiti del Supporto Decisionale Locale	21
5.2	Modellazione Matematica delle Serie Temporali Fisiologiche	21
5.2.1	Media Mobile Esponenziale Ponderata (EMA)	21
5.2.2	Stima della Deriva Temporale (Trend Slope OLS)	22
5.3	Implementazione dell'Actor di Supporto Decisionale in Swift	22
5.4	Implementazione dei Protocolli Clinici Internazionali	23
5.4.1	Valutazione del Profilo Pressorio (Linee Guida ESC/ESH 2023)	23
5.4.2	Valutazione del Profilo Glicemico e Metabolico (Linee Guida ADA 2024)	24
5.5	Generazione di Report Aggregati per il Medico di Medicina Generale (MMG)	24

6	Ingegneria del Software ed Ecosistema Apple	25
6.1	Swift Concurrency e Isolamento con Actor	25
6.2	Persistenza Reattiva con SwiftData e @ModelActor	25
6.3	Architettura UI Reattiva con SwiftUI e Decomposizione Atomica	26
6.3.1	Theme Engine Clinico Dinamico	27
6.4	Accessibilità Universale (VoiceOver & Semantica Grafica)	27
6.5	Ecosistema watchOS: Glanceability e Sincronizzazione Locale	27
6.6	Share Extension: Ingestione Documentale Zero-Click	28
7	Valutazione Sperimentale, Test Suite e Benchmark	28
7.1	Metodologia di Sviluppo TDD e Architettura Multi-Tier della Test Suite	28
7.2	Test di Robustezza su Input Avversari (Adversarial Security & Fuzzing)	29
7.3	Benchmark Prestazionali su Hardware Fisico Reale e Protocollo Sperimentale	29
7.3.1	Harness di Misurazione e Protocollo di Riproducibilità	29
7.3.2	Stratificazione dei Payload Sperimentali	30
7.4	Profilazione Energetica e Consumo Batteria (Instruments Energy Trace)	30
7.5	Accuratezza della Transcodifica Semantica e Metodologia di Validazione	30
7.5.1	Composizione del Corpus Sperimentale e Considerazioni Deontologiche	31
7.5.2	Risultati Sperimentali e Analisi Critica delle Metriche	31
8	Conclusioni, EHDS e Sviluppi Futuri	32
8.1	Sintesi dei Risultati e Impatto Tecnologico	32
8.2	Limiti Attuali dell'Implementazione	32
8.3	Prospettive Evolutive e Roadmap di Ricerca (Stato dell'Arte vs Proposte)	33
8.3.1	Convergenza verso lo Spazio Europeo dei Dati Sanitari (EHDS)	33
8.3.2	Esplorazione di Small Language Models (SLM) On-Device (Proposta di Ricerca)	33
8.4	Conclusioni	34
A	Catalogo Ontologico di Transcodifica LOINC → Apple HealthKit e Vault FHIR	35
A.1	Binario 1: Segni Vitali e Parametri Antropometrici (HKQuantitySample)	35
A.2	Binario 2: Chimica Clinica ed Ematologia (Vault FHIR e Record Clinici)	36
B	Payload XML HL7 CDA R2 Esempio e Annotato	37
C	Specifica Ingegneristica XcodeGen (project.yml)	39
D	Profilo di Attribuzione Accademica, Identificatori Persistenti e Istruzioni di Citazione	41
D.1	Dati Istituzionali dell'Autore	41
D.2	Guida alla Citazione Bibliografica	41
D.2.1	Formato BibTeX	41
D.2.2	Formato APA (7th Edition)	41
D.2.3	Formato IEEE	42
D.3	Dichiarazione sull'Accesso Aperto e Proprietà Intellettuale	42

1 Introduzione e Inquadramento del Problema

1.1 Il Paradosso della Sanità Digitale e i Silos Informativi

Nel corso dell'ultimo decennio, i sistemi sanitari dei paesi industrializzati hanno intrapreso una massiccia transizione digitale, volta a dematerializzare la documentazione clinica cartacea e ad accentrare la storia medica dei pazienti all'interno di architetture regionali e nazionali. In Italia, tale processo ha trovato la sua massima espressione nell'evoluzione del **Fascicolo Sanitario Elettronico (FSE)**, culminata recentemente nel programma strategico **FSE 2.0** promosso dal Ministero della Salute e dal Dipartimento per la Trasformazione Digitale nell'alveo delle risorse del Piano Nazionale di Ripresa e Resilienza (PNRR) [1].

Nonostante gli ingenti investimenti infrastrutturali e la disponibilità di canali di accesso telematici protetti (quali SPID e Carta d'Identità Elettronica), emerge un vistoso *paradosso dell'interoperabilità*: la dematerializzazione non si è tradotta in una reale emancipazione computazionale del dato medico a beneficio del cittadino. Nella stragrande maggioranza dei contesti ospedalieri e diagnostici territoriali, il referto specialistico o il referto di laboratorio (esami ematochimici, microbiologici, citologici) viene recapitato al paziente sotto forma di documento statico in formato **PDF** (Portable Document Format) o, nei casi tecnologicamente più avanzati, quale payload XML conforme al profilo architeturale **HL7 CDA Release 2** (Clinical Document Architecture) [2].

Dal punto di vista dell'assistito, il formato PDF non strutturato impedisce l'elaborazione analitica automatizzata e l'integrazione semantica continua (*data tomb*):

1. **Assenza di granularità analitica:** I valori numerici di esami vitali (glicemia basale, emoglobina glicata, colesterolemia LDL, filtrato glomerulare stimato, creatinina) rimangono imprigionati in una matrice grafica destinata esclusivamente alla lettura visiva o alla stampa, impedendo qualunque forma di tracciamento temporale continuativo.
2. **Invisibilità longitudinale:** Il paziente cronico o multimorboso (affetto da sindrome metabolica, ipertensione o diabete di tipo 2) riceve decine di referti isolati nell'arco degli anni. Ciascun documento rappresenta un'istantanea puntiforme priva di collegamenti con i rilievi precedenti, rendendo difficoltosa la visualizzazione di derive patologiche lente o di miglioramenti terapeutici.
3. **Frattura rispetto agli ecosistemi di monitoraggio consumer:** Contestualmente all'evoluzione dell'FSE, si è assistito a una pervasiva proliferazione di dispositivi indossabili di grado consumer o medicale (*smartwatch*, fasce cardio, glucometri continui CGM, sfingomanometri wireless). Framework operativi moderni come **Apple HealthKit** [3] raccolgono quotidianamente miliardi di punti dati fisiologici ad alta frequenza (frequenza cardiaca a riposo, variabilità HRV, stime VO2 max, monitoraggio del sonno). Tuttavia, questo ricco ecosistema consumer vive in un isolamento totale rispetto ai dati clinici certificati emessi dai laboratori ospedalieri.

1.2 La Sovranità del Dato e l'Imperativo Zero-Cloud

Nel tentativo di colmare questo divario, molteplici aziende del settore *mHealth* (Mobile Health) hanno sviluppato applicazioni capaci di ingerire referti medici caricandoli su piattaforme cloud proprietarie. Tale approccio, sebbene apparentemente conveniente sotto il profilo dello sviluppo software, introduce rischi inaccettabili in termini di sicurezza e riservatezza:

- **Esposizione di categorie particolari di dati (GDPR Art. 9):** I dati relativi alla salute, per loro natura intrinseca, non sono revocabili né modificabili a seguito di un data breach. Il caricamento di referti ematochimici completi su server terzi moltiplica la superficie

d'attacco ed espone l'utente a rischi sistemici di esfiltrazione, profilazione assicurativa o tracciamento pubblicitario non autorizzato [4].

- **Vulnerabilità da Vendor Lock-in e Interruzione del Servizio:** Se il provider cloud cessa l'attività, modifica le proprie condizioni economiche o subisce un disservizio di rete, il paziente perde l'accesso immediato alla propria cartella clinica digitale aggregata.

Sulla scorta di queste premesse, emerge con forza la necessità di un paradigma architetture divergente: l'approccio **Zero-Cloud**. In un sistema Zero-Cloud, l'intero ciclo di vita del dato clinico (acquisizione, decrittazione, validazione semantica, parsing dell'albero XML, transcodifica ontologica, inferenza decisionale e memorizzazione a riposo) deve essere eseguito **esclusivamente sul dispositivo fisico del paziente (edge/on-device computing)** [5], senza che alcun singolo byte varchi il perimetro di rete dello smartphone.

1.3 Obiettivi e Contributi del Progetto RosettaEdge

Il progetto **RosettaEdge** nasce per colmare in modo strutturato il divario tra la sanità istituzionale e il monitoraggio personale del cittadino. L'applicazione si configura come un motore di mediazione semantica (*Semantic Mediation Engine*) on-device sviluppato nativamente in linguaggio Swift per l'ecosistema operativo Apple (iOS e watchOS).

I contributi originali formalizzati all'interno di questo lavoro comprendono:

1. **Architettura Edge Deterministica e Zero-Cloud:** Dimostrazione della fattibilità di un'architettura di sanità digitale operante interamente sul dispositivo senza componenti server-side o cloud esterni, orientata ai principi di *Privacy-by-Design* e *Privacy-by-Default* sanciti dal Regolamento Europeo 2016/679 (GDPR) [6].
2. **Il Rosetta Transcoder Engine:** Un sottosistema di parsing XML in streaming con consumo di memoria limitato per documenti conformi al profilo nazionale HL7 CDA Release 2 del Fascicolo Sanitario Elettronico italiano, integrato con una matrice di transcodifica ontologica deterministica tra i codici di laboratorio **LOINC** [7], le grandezze metriche **UCUM** [8] e gli identificatori tipizzati di **Apple HealthKit** (`HKQuantityType`) per i parametri supportati.
3. **Identity Firewall e Prevenzione della Contaminazione:** Progettazione di un filtro architetture di verifica dell'identità on-device che previene la contaminazione del database personale di Apple Salute con referti appartenenti a congiunti o terzi (caso d'uso frequente nella gestione dei familiari da parte dei caregiver), fondata sulla combinazione di validazione algebrica del Codice Fiscale ed euristiche di string matching (distanza di Jaro-Winkler) [9].
4. **Pipeline OCR con Validazione Human-in-the-Loop:** Sviluppo di un modulo di ripiego (*fallback*) per referti analogici o PDF rasterizzati basato sul framework on-device Apple Vision, governato da un protocollo a conferma esplicita per impedire l'iniezione non supervisionata di dati biometrici errati nell'archivio clinico di sistema.
5. **Supporto Decisionale Clinico (CDS) Deterministico On-Device:** Implementazione di algoritmi analitici longitudinali per la valutazione delle serie storiche cardiovascolari e glicemiche, parametrizzati secondo le linee guida internazionali delle società scientifiche ESC/ESH (per l'ipertensione arteriosa) [10] e ADA (per il diabete mellito) [11], intesi come strumento di consultazione e non come dispositivo medico sostitutivo del parere clinico.
6. **Protezione dei Dati a Riposo e Controllo Biometrico:** Adozione della classe di protezione di sistema `NSFileProtectionComplete` di iOS per l'archivio locale SQLite/SwiftData, combinata con il controllo degli accessi biometrici (Face ID o Touch ID) vincolato a chiavi custodite nel Secure Enclave tramite Keychain Services (`kSecAccessControlBiometryAny`) [12].

1.4 Struttura della Monografia

Il presente documento è articolato secondo la seguente struttura logica:

- La **Section 2** analizza lo stato dell'arte dell'interoperabilità sanitaria, dettagliando le specifiche dello standard HL7 CDA R2, il panorama del FSE 2.0 in Italia, le ontologie mediche di riferimento (LOINC, UCUM) e i framework di salute mobile.
- La **Section 3** espone l'architettura complessiva di RosettaEdge, il modello di minaccia formale (STRIDE), l'analisi di conformità al GDPR e la gestione della crittografia con Secure Enclave.
- La **Section 4** descrive in profondità il motore *Rosetta Engine*, formalizzando il parsing XML, la mappatura semantica verso HealthKit, l'algoritmo di *Identity Firewall* e il modulo Vision OCR.
- La **Section 5** formalizza gli algoritmi matematici di *Clinical Decision Support* per l'analisi longitudinale dei trend e la generazione di report clinici aggregati.
- La **Section 6** illustra le scelte implementative relative a Swift 5.10/6, Swift Concurrency, SwiftUI, architettura del workspace XcodeGen e la companion app per watchOS.
- La **Section 7** presenta i risultati sperimentali, la copertura dei test (TDD), la resilienza a input avversari e i benchmark prestazionali di latenza e consumo energetico eseguiti su hardware reale.
- La **Section 8** conclude la trattazione analizzando le prospettive evolutive connesse allo Spazio Europeo dei Dati Sanitari (EHDS) e all'integrazione di Small Language Models (SLM) on-device.

2 Stato dell'Arte e Fondamenti Teorici

2.1 L'Evoluzione degli Standard di Interoperabilità Sanitaria

L'interoperabilità informatica in sanità è comunemente classificata su quattro livelli gerarchici: *tecnica* (trasmissione di flussi di bit), *sintattica* (strutturazione formale dei dati), *semantica* (condivisione non ambigua del significato clinico) e *organizzativa* (armonizzazione dei processi di cura) [13].

Negli ultimi tre decenni, l'organizzazione internazionale **Health Level Seven (HL7)** ha guidato l'evoluzione dei modelli di interoperabilità sintattica e semantica:

- **HL7 Version 2 (v2.x)**: Standard basato su messaggistica orientata a eventi, serializzato tramite una sintassi proprietaria delimitata da pipe (|) e cappelletti (^). Pur essendo tuttora pervasivo all'interno delle reti locali ospedaliere per la comunicazione tra Laboratory Information Systems (LIS) e Hospital Information Systems (HIS), HL7 v2 difetta di una semantica clinica formale e vincolante, delegando ampie porzioni di significato a campi opzionali (*Z-segments*) che ne frammentano l'uniformità.
- **HL7 Clinical Document Architecture (CDA Release 2)**: Standard ratificato da ANSI e standardizzato come ISO/HL7 27932:2009 [2]. Il CDA R2 segna il passaggio dal concetto di "messaggio effimero" a quello di "documento clinico persistente", strutturato in formato XML e fondato sul Reference Information Model (RIM) di HL7 v3.
- **HL7 Fast Healthcare Interoperability Resources (FHIR R4)**: Standard moderno orientato alle architetture web, che scompone le entità cliniche in "Risorse" modulari ed estensibili (*Patient, Observation, DiagnosticReport, Encounter*), serializzabili sia in

JSON che in XML e accessibili tramite paradigmi RESTful e protocolli di autorizzazione SMART on FHIR (OAuth 2.0 / OIDC) [14]. Nonostante FHIR rappresenti la direzione strategica verso cui converge la sanità internazionale, la sua adozione nelle strutture ospedaliere italiane territoriali è tuttora marginale o in fase embrionale, lasciando il CDA R2 quale unico standard vincolante di produzione reale dei referti.

2.2 Anatomia del Documento HL7 CDA Release 2

Un documento conforme a HL7 CDA R2 è un oggetto XML auto-contenuto, clinicamente persistente, legalmente autenticabile e strutturato in due macro-componenti fondamentali (Figure 1):

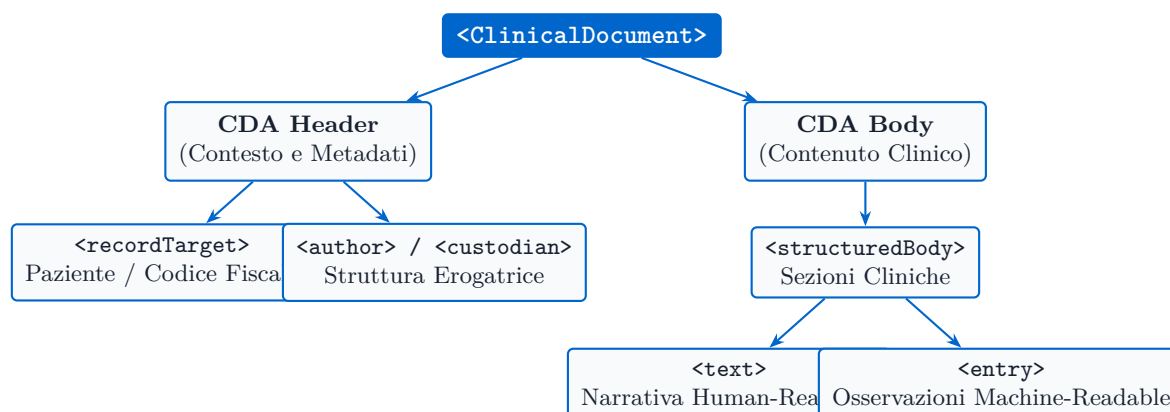


Figura 1: Albero gerarchico concettuale di un documento clinico HL7 CDA Release 2.

1. **CDA Header (Intestazione):** Definisce il contesto amministrativo, legale e anagrafico del documento. I blocchi chiave includono:

- **recordTarget:** Identifica il soggetto a cui si riferisce la prestazione sanitaria. Include l'elemento **patientRole** che ospita l'identificativo fiscale nazionale (Codice Fiscale, rappresentato tramite OID di namespace 2.16.840.1.113883.2.9.4.3.2), i dati anagrafici e la residenza.
- **author** e **legalAuthenticator:** Documentano il professionista sanitario o il firmatario legale che assume la responsabilità clinica del contenuto.
- **custodian:** Specifica la struttura sanitaria, l'Azienda Sanitaria Locale (ASL) o l'Azienda Ospedaliero-Universitaria custode dell'archivio digitale.

2. **CDA Body (Corpo del Referto):** Ospita il contenuto clinico ed è organizzato all'interno del nodo **structuredBody**. Il body è suddiviso in **section** tematiche (es. Chimica Clinica, Ematologia, Coagulazione), ciascuna comprendente:

- Una componente narrativa human-readable all'interno del tag **text** (solitamente formattata in blocchi HTML semplificati contenenti tabelle).
- Una componente strutturata machine-readable racchiusa in blocchi **entry**, contenenti nodi **observation** tipizzati con attributo **classCode="OBS"** e **moodCode="EVN"**.

Nel listato seguente viene riportata la serializzazione canonica di un'osservazione di laboratorio nel profilo italiano:

```

1 <entry typeCode="DRIV">
2   <observation classCode="OBS" moodCode="EVN">
3     <templateId root="2.16.840.1.113883.2.9.10.1.11"/>

```

```

4   <id root="2.16.840.1.113883.2.9.2.120.4.4" extension="OBS-2026-1029"/>
5   <!-- Codice LOINC standard per glucosio basale -->
6   <code code="1558-6" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
7   ↪ displayName="Fasting Glucose [Mass/volume] in Serum or Plasma"/>
8   <statusCode code="completed"/>
9   <effectiveTime value="20261020083000+0200"/>
10  <!-- Grandezza fisica (Physical Quantity) con unita UCUM -->
11  <value xsi:type="PQ" value="94" unit="mg/dL"/>
12  <interpretationCode code="N" codeSystem="2.16.840.1.113883.5.83"/>
13  <referenceRange typeCode="REFV">
14    <observationRange classCode="OBS" moodCode="EVN.CRT">
15      <value xsi:type="IVL_PQ">
16        <low value="70" unit="mg/dL"/>
17        <high value="99" unit="mg/dL"/>
18      </value>
19    </observationRange>
20  </referenceRange>
21 </entry>

```

Listing 1: Esempio di osservazione strutturata HL7 CDA R2 per la glicemia a digiuno.

2.3 L’Infrastruttura Nazionale di Interoperabilità (INI) e i Sistemi Regionali

In Italia, la trasformazione digitale della sanità ha subito una decisiva accelerazione con il Piano Nazionale di Ripresa e Resilienza (PNRR, Missione 6 Salute, Componente 2, Sub-investimento 1.3.1 “Fascicolo Sanitario Elettronico”), disciplinato dal **Decreto 29 settembre 2023** emanato dal Ministero della Salute di concerto con il Dipartimento per la Trasformazione Digitale e il MEF [1]. Tale regolamento istituisce l’**Infrastruttura Nazionale per l’Interoperabilità (INI)**, affidata tecnologicamente a Sogei e coordinata operativamente da AGENAS.

L’infrastruttura si articola su due livelli cooperativi:

1. **Il Gateway Nazionale INI:** Funge da broker di indicizzazione (*Registry*) e instradamento cross-regionale, consentendo a un medico di una determinata regione di consultare documenti sanitari generati da strutture sanitarie di un’altra regione.
2. **I Repository Regionali:** Custodiscono fisicamente i documenti clinici CDA firmati digitalmente in formato XAdES/CAAdES. L’eterogeneità storica delle infrastrutture regionali ha dato origine a tre modelli architetturali prevalenti:
 - **Modello IHE-XDS.b (Regione Emilia-Romagna):** Il sistema *FSE-ER* adotta rigorosamente il profilo *Cross-Enterprise Document Sharing* (IHE XDS.b) basato su Web Services SOAP conformi a WS-Security e SAML 2.0. I documenti HL7 CDA R2 sono archiviati in repository distribuiti facenti capo alle diverse Aziende USL e Ospedaliero-Universitarie.
 - **Modello Service-Oriented (Regione Lombardia):** Il Fascicolo Sanitario Elettronico lombardo poggia su un gateway denominato *GSA* (Gestione Servizi di Accesso) che espone API RESTful/SOAP per l’interrogazione federata delle cartelle cliniche elettroniche delle ASST territoriali.
 - **Modello Sanità km zero (Regione Veneto):** Piattaforma centralizzata nativa digitale che unifica la ricetta dematerializzata, i referti ospedalieri e le prenotazioni specialistiche, basata su autenticazione SPID/CIE di livello 2 e notifiche push al cittadino.

Nonostante l’eccellenza ingegneristica dell’infrastruttura di backend, l’ultimo miglio (*last mile*) tra il sistema sanitario e il paziente soffre di un marcato divario di usabilità: i portali regionali convertono i file CDA strutturati in rendering PDF statici al solo fine di visualizzazione grafica, precludendo l’importazione automatica delle serie temporali fisiologiche nei dispositivi mobili dei cittadini.

2.4 Ontologie e Sistemi di Codifica Clinica

Affinché un'applicazione software possa interpretare in modo non ambiguo una rilevazione medica, il dato deve essere ancorato a vocabolari controllati standardizzati a livello internazionale:

2.4.1 LOINC (Logical Observation Identifiers Names and Codes)

Sviluppato dal Regenstrief Institute [7], LOINC è il sistema universale per l'identificazione univoca delle osservazioni di laboratorio e delle misurazioni cliniche. Ogni codice numerico (es. 1558-6 per il glucosio nel siero/plasma) identifica univocamente sei assi fondamentali:

- *Componente/Analita misurato* (es. Glucosio, Colesterolo HDL)
- *Proprietà misurata* (es. Massa/Volume, Concentrazione di sostanza)
- *Aspetto temporale* (es. Puntiforme Pt vs Raccolta 24 ore 24h)
- *Sistema/Matrice biologica* (es. Siero, Plasma, Sangue intero, Urine)
- *Tipo di scala* (es. Quantitativa Qn, Nominale Nom)
- *Metodo di analisi* (opzionale, specificato solo se clinicamente discriminante)

All'interno dei documenti CDA, il namespace LOINC è formalmente identificato dall'OID numerico 2.16.840.1.113883.6.1.

2.4.2 UCUM (Unified Code for Units of Measure)

Le misurazioni quantitative di tipo fisico (PQ) all'interno di un documento CDA necessitano di un'indicazione rigorosa dell'unità di misura. Lo standard UCUM [8] fornisce una sintassi formale e non ambigua, rigorosamente case-sensitive, per esprimere tutte le grandezze fisiche del Sistema Internazionale (SI) e le unità d'uso clinico tradizionale (mg/dL, mmol/L, mm[Hg]).

2.4.3 ICD (International Classification of Diseases)

Per la codifica delle diagnosi patologiche e delle cause di ricovero, gli standard di riferimento sono l'ICD-9-CM e l'ICD-10 / ICD-11 promosso dall'Organizzazione Mondiale della Sanità (OMS).

2.5 Ecosistemi di Mobile Health: Apple HealthKit vs Google Health Connect

Sul versante dei sistemi operativi mobili consumer, le due piattaforme di riferimento hanno sviluppato architetture specializzate per la centralizzazione dei parametri fisiologici:

- **Apple HealthKit:** Introdotto con iOS 8 e integrato in watchOS [3], funge da archivio centralizzato crittografato sul dispositivo. Offre un modello di autorizzazione asimmetrico e granulare (`HKQuantityTypeIdentifier`), garantendo che ogni tipo di dato debba essere autorizzato esplicitamente. Supporta sia campioni quantitativi continui (`HKQuantitySample`), sia documenti complessi strutturati (`HKCDA DocumentSample`).
- **Google Health Connect:** Soluzione introdotta su Android per federare i dati biometrici tra app di terze parti mediante un archivio on-device con controlli di accesso granulari, orientata principalmente a fitness e tracciamento dell'attività motoria.

2.6 Analisi Comparativa Multidimensionale degli Standard

La [Table 1](#) sintetizza il confronto analitico tra i principali standard e framework esaminati in relazione ai requisiti di un'architettura mobile decentralizzata.

Tabella 1: Confronto multidimensionale tra standard di interoperabilità clinica e framework mHealth.

Dimensione	HL7 v2.x	HL7 CDA R2	HL7 FHIR R4	Apple HealthKit	Google Health Connect
Paradigma Base	Messaggistica eventi	a Documento persistente	Risorse RESTful atomiche	Archivio locale on-device	Storage locale federato
Formato Dati	Testo delimitato (I)	XML conforme a RIM	JSON / XML	Oggetti nativi Swift/Obj-C	Jetpack Data Layer (Kotlin)
Semantica Clinica	Debole (Z-segments)	Rigorosa (LOINC/UCUM)	Rigorosa (SNO-MED/LOINC)	Rigorosa circoscritta	Fitness-oriented / Base
Validità Legale FSE	Nulla (Uso interno)	Piena (Firma XAdES)	In divenire (EEHRxF)	Nessuna (Uso personale)	Nessuna (Uso personale)
Edge Mobile Efficiency	Bassa (Parsing manuale)	Elevata con streaming SAX	Elevata con JSON streaming	Massima (Hardware API)	Elevata (Android Jetpack)
Cifratura a Riposo	Delegata all'infrastruttura	Delegata all'archivio	TLS / HTTPS a riposo	Hardware Secure Enclave	Android Keystore / File-Based
Modello Privacy	B2B Ospedaliero	B2B / G2G Istituzionale	OAuth 2.0 / Scopes	Granulare / metrico	Asimmetrica / Autorizzazione per Permessi

3 Architettura di Sistema e Modello di Sicurezza Zero-Cloud

3.1 Il Paradigma Zero-Cloud e l'Edge Computing Sanitario

L'architettura software di **RosettaEdge** adotta un paradigma computazionale mobile-edge alternativo al modello client-server convenzionale che governa la maggior parte delle applicazioni sanitarie contemporanee. Nel modello convenzionale, il terminale mobile funge da mero visualizzatore (*thin client*), mentre le operazioni computazionalmente onerose (parsing documentale, estrazione di entità con modelli di Natural Language Processing, analytics longitudinale) vengono delegate a cluster remoti nel cloud.

Al contrario, RosettaEdge adotta un paradigma **Mobile Edge Computing puro**, in cui l'intero ciclo di elaborazione e conservazione è circoscritto all'interno della sandbox del sistema operativo host (iOS). Come illustrato nella [Figure 2](#), l'architettura è strutturata secondo un modello modulare di tipo **Producer-Consumer disaccoppiato**:

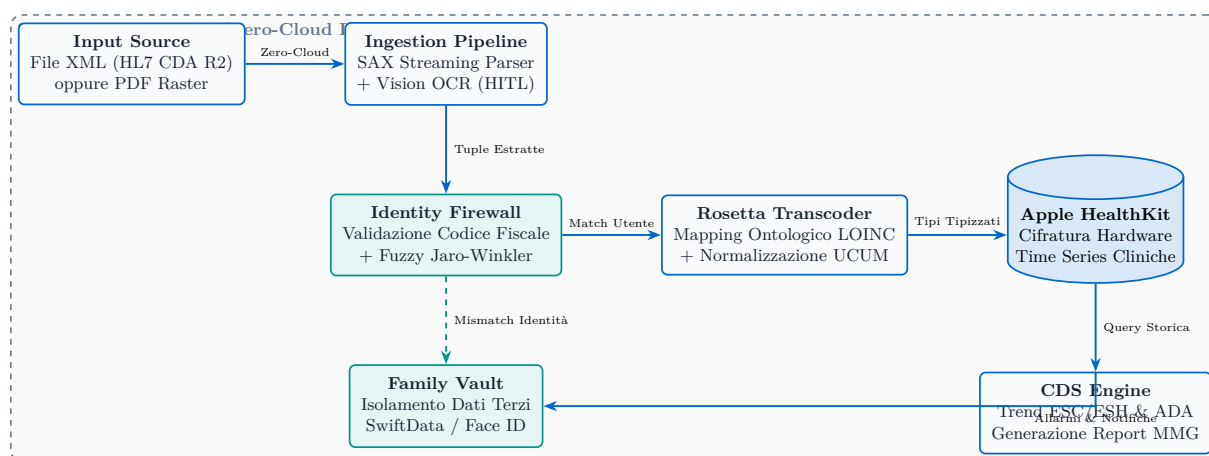


Figura 2: Diagramma architetturale del flusso dati di RosettaEdge: modello Producer-Consumer rigorosamente confinato nella sandbox locale del dispositivo mobile.

I componenti fondamentali dell'architettura si articolano come segue:

1. **Data Ingestion Layer (Producer):** Intercetta i file clinici attraverso la Share Sheet di sistema, l'apertura da app esterne (Mail, Safari, File) o l'acquisizione tramite fotocamera/scanner integrato.

2. **Verification & Firewall Layer:** Esegue il parsing deterministico in streaming, verifica l'identità del titolare del referto tramite l'Identity Firewall e instrada il payload verso il database personale di salute o verso il vault protetto per i familiari.
3. **Transcoding & Normalization Core:** Esegue la conversione ontologica tra le codifiche LOINC di laboratorio e le entità tipizzate di Apple HealthKit, applicando le formule di normalizzazione dimensionale conformi allo standard UCUM.
4. **Secure Storage & Clinical Analytics Layer (Consumer):** Scrive i campioni quantitativi in Apple HealthKit, persiste i metadati e i documenti originali in SwiftData cifrato con Secure Enclave, ed alimenta il motore di Clinical Decision Support (CDS) per il monitoraggio dei trend cronici.

3.2 Analisi di Conformità al Regolamento Generale sulla Protezione dei Dati (GDPR)

Nel trattamento dei dati relativi alla salute, la quasi totalità delle applicazioni commerciali si scontra con la complessità del **Regolamento (UE) 2016/679 (GDPR)**. Ai sensi dell'**Articolo 9, paragrafo 1**, il trattamento delle cosiddette “categorie particolari di dati” (tra cui i dati genetici, biometrici e relativi alla salute) è generalmente vietato, salvo la sussistenza di specifiche deroghe (es. consenso esplicito informato, finalità di medicina preventiva o diagnosi, interesse pubblico nel settore sanitario).

Nel contesto di RosettaEdge, la conformità non è considerata una conseguenza passiva dell'assenza di server remoti, bensì il risultato di un'architettura proattivamente allineata ai requisiti di **Privacy-by-Design e Privacy-by-Default (Articolo 25)** e alle misure tecniche di sicurezza adeguate al rischio (**Articolo 32**):

1. **Minimizzazione e Assenza di Trattamento Centrale:** Non sussiste alcuna infrastruttura server, cloud intermedio o database relazionale remoto gestito dallo sviluppatore o da terzi:

$$\forall d \in \mathcal{D}_{\text{clinici}}, \quad \text{NetworkTraffic}(d) = \emptyset \quad (1)$$

Ciò esclude la necessità di designare Responsabili del Trattamento (Art. 28) per l'archiviazione dei referti e riduce drasticamente la superficie d'attacco rispetto a soluzioni centralizzate, eliminando l'esposizione a violazioni massive dei dati (*data breach*) su scala aggregata.

2. **Sovranità e Gestione Individuale del Dato:** Il cittadino-paziente assume il controllo esclusivo del dato clinico importato nel proprio dispositivo personale. Conformemente agli orientamenti dell'*European Data Protection Board* (EDPB) sulle applicazioni sanitarie mobili, il trattamento locale risponde a una finalità strettamente individuale di consultazione, gestione della salute personale e integrazione terapeutica con il proprio medico curante.
3. **Assenza di Tracciamento e Telemetria:** L'applicativo è privo di framework analitici, SDK di telemetria remota o identificatori pubblicitari (IDFA). Nessun metadato diagnostico o profilo comportamentale abbandona l'ecosistema del terminale.
4. **Sincronizzazione HealthKit Regolata dall'Utente:** L'eventuale propagazione dei campioni verso Apple HealthKit o il backup cloud di sistema (iCloud) non è gestita dall'app ma dalle preferenze di sistema operativo dell'utente, che operano sotto crittografia end-to-end con protezione avanzata dei dati (ADP) abilitabile dall'interessato.

3.2.1 Valutazione d’Impatto sulla Protezione dei Dati (DPIA)

A tutela dell’utente, l’applicazione integra un framework rigoroso di **Data Protection Impact Assessment (DPIA)** pre-implementativo. Tra i presidi di garanzia implementati:

- **Minimizzazione del Dato (Art. 5, par. 1, lett. c):** Vengono estratti e persistiti unicamente i parametri biometrici indispensabili per la diagnostica (valore numerico, unità, data e ora, intervallo di normalità). Note amministrative o anagrafiche non correlate vengono scartate dopo l’ispezione.
- **Diritto all’Oblio e Cancellazione Fisica (Art. 17):** L’interfaccia utente include comandi distruttivi fisici (`.onDelete`). La cancellazione di un’osservazione o di un referto elimina istantaneamente il record dallo storage SwiftData locale e provvede alla propagazione del comando di cancellazione verso il database HealthKit (`deleteObject`). Nessun dato permane in tabelle di log o backup remoti.

3.3 Modellazione delle Minacce (STRIDE Threat Model)

La sicurezza di un’applicazione sanitaria mobile priva di backend deve essere fondata sulla mitigazione degli attacchi locali (furto del dispositivo, estrazione forense, malware presente sull’host, librerie malevole di terze parti). È stato condotto un’analisi formale delle minacce secondo la metodologia **STRIDE** elaborata da Microsoft, i cui risultati e contromisure sono sintetizzati nella [Table 2](#).

Tabella 2: Modello di minaccia STRIDE applicato all’architettura di RosettaEdge.

Categoria Minaccia	Vettore di Rischio Specifico	Meccanismo di Mitigazione Implementato
Spoofing (Falsificazione)	Un utente importa un referto clinico alterato o appartenente a un’altra persona per falsificare il proprio profilo HealthKit.	Identity Firewall: Verifica del Codice Fiscale estratto e calcolo della distanza di Jaro-Winkler sul nominativo rispetto al profilo locale.
Tampering (Manomissione)	Modifica dei parametri ematochimici a riposo nel file di database locale da parte di un aggressore fisico.	Database cifrato tramite Data Protection API di iOS (<code>NSFileProtectionComplete</code>) e digest HMAC custodito in Keychain.
Repudiation (Ripudio)	Disconoscimento della fonte di una misurazione inserita nel grafico clinico.	Ogni campione iniettato in HealthKit conserva metadati di provenienza immutabili (<code>HKMetadataKeyDevice</code> , hash sorgente CDA).
Information Disclosure (Fuga di Dati)	Ispezione della memoria RAM o accesso ai referti da parte di congiunti che usano il telefono sbloccato.	Biometric Gating: Schermata di blocco applicativo con Face ID/Touch ID; svuotamento dei buffer XML non appena concluso il parsing.
Denial of Service (Disponibilità)	Upload di file XML maligni costruiti per saturare la RAM (attacchi <i>Billion Laughs</i> o Entity Expansion).	Disattivazione esplicita della risoluzione di DTD/entità esterne nel parser <code>XMLParser</code> nativo di Apple; limiti rigidi di dimensione file.
Elevation of Privilege (Privilegi)	Accesso ai dati clinici da parte di app terze prive di autorizzazione HealthKit.	Isolamento delle sandbox iOS e gestione rigorosa delle autorizzazioni HealthKit gestite a livello kernel da Apple.

3.4 Architettura di Sicurezza Hardware: Apple Secure Enclave e Gerarchia delle Chiavi

Per garantire la riservatezza e la protezione dei dati a riposo nel caso in cui il dispositivo fisico venga smarrito o sottratto, RosettaEdge sfrutta l'architettura crittografica hardware integrata nei dispositivi Apple [12], schematizzata nella [Figure 3](#).

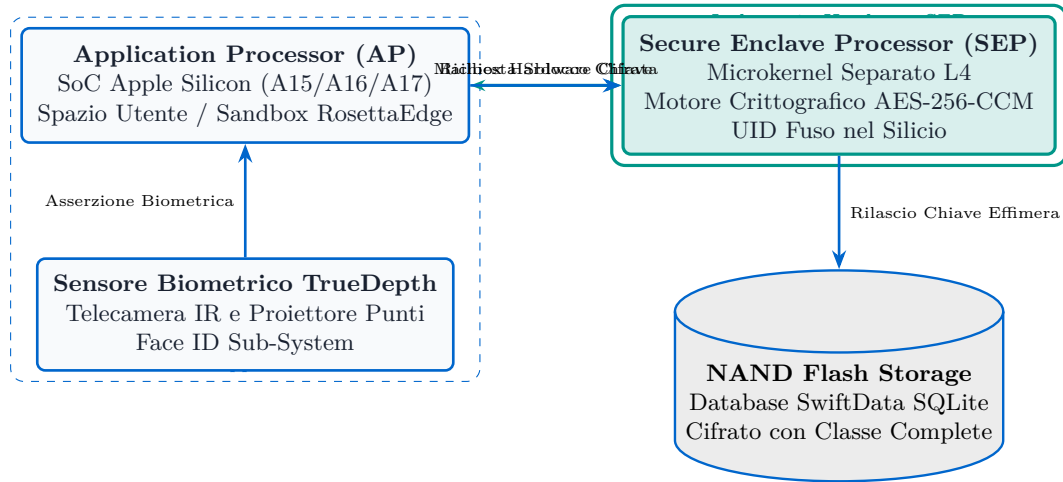


Figura 3: Isolamento hardware tra Application Processor e Secure Enclave Processor.

3.4.1 Apple Keychain Services e Secure Enclave

Il Codice Fiscale dell'utente e le credenziali sensibili non vengono mai serializzati su file di testo o database convenzionali. Essi risiedono all'interno del **Keychain di sistema**, con attributi di sicurezza massimi:

- **kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly**: Garantisce che il dato sia accessibile unicamente a dispositivo sbloccato e impedisce il trasferimento del payload crittografico tramite backup iCloud o migrazione via cavo su un nuovo terminale.
- **SecAccessControlCreateWithFlags**: La chiave viene protetta da una condizione logica hardware gestita dal coprocessore **Secure Enclave**:

$$ACL = kSecAccessControlBiometryAny \vee kSecAccessControlUserPresence \quad (2)$$

Ogni tentativo di recupero della chiave per decifrare l'identità del proprietario richiede fisicamente la convalida biometrica dell'utente tramite Face ID o Touch ID.

3.4.2 SwiftData e Data Protection API

La persistenza dei metadati clinici locali è delegata a **SwiftData** [15]. L'architettura di sicurezza di RosettaEdge adotta una rigorosa separazione su due livelli distinti per la protezione dei dati:

1. **Protezione del Filesystem Locale (Data Protection Classe A)**: Il database sottostante SQLite (gestito tramite SwiftData/Core Data) è configurato con la classe di protezione **NSFileProtectionComplete**. La chiave perimetrale di cifratura del file viene derivata combinando l'UID hardware del dispositivo con il codice di sblocco dell'utente; non appena il dispositivo entra in stato di blocco schermo (*screen lock*), il kernel di iOS espunge la chiave di classe dalla memoria RAM. Ciò garantisce una solida mitigazione forense a riposo contro attacchi fisici BFU (*Before First Unlock*) e AFU (*After First Unlock*) a terminale bloccato.

2. **Controllo degli Accessi Hardware-Bound (Secure Enclave):** I segreti crittografici ad alta sensibilità (inclusi i salt per l'Identity Firewall e le chiavi del Family Vault) non risiedono nel database generale, ma vengono gestiti tramite Keychain Services con vincolo hardware `kSecAccessControlBiometryAny` e flag `kSecAttrAccessibleWhenUnlockedThisDeviceOnly`. Ogni operazione crittografica richiede l'autorizzazione fisica del coprocessore Secure Enclave Processor (SEP) subordinata alla verifica biometrica positiva (Face ID/Touch ID), isolando i vettori d'attacco dall'Application Processor (AP).

4 Il Motore di Transcodifica Semantica (Rosetta Engine)

4.1 Architettura del Parser XML: Streaming SAX vs Document Object Model

L'elaborazione di documenti clinici su dispositivi mobili pone vincoli stringenti in termini di impronta di memoria RAM (*memory footprint*) e latenza di esecuzione. Un referto ospedaliero completo emesso secondo lo standard HL7 CDA R2 (ad esempio una cartella di laboratorio comprendente emocromo con formula, assetto lipidico, coagulazione e profilo renale) genera payload XML compresi tra 150 KB e oltre 2 MB di testo marcato.

In contesti informatici tradizionali, il parsing di file XML viene frequentemente affidato a parser basati su **DOM (Document Object Model)**, i quali istanziano in memoria un albero di nodi completo prima di consentirne l'interrogazione tramite espressioni XPath. Nei dispositivi mobili, tuttavia, l'uso di un parser DOM presenta vulnerabilità critiche:

- **Overhead di Memoria:** La generazione dell'albero gerarchico DOM comporta una moltiplicazione dell'occupazione di memoria variabile da 5 a 10 volte la dimensione del file grezzo, innescando potenziali crash dell'app per esaurimento della memoria (terminazione da parte del demone *Jetsam* del kernel XNU di iOS).
- **Vulnerabilità ad Attacchi di Espansione XML:** I parser DOM sono storicamente suscettibili a denial-of-service basati su entità ricorsive (quali il noto attacco *Billion Laughs* o attacchi XXE - XML External Entity).

Per superare tali limitazioni, il **Rosetta Engine** implementa un'architettura ibrida fondata su un parser a eventi di tipo **SAX (Simple API for XML)** incapsulato all'interno della classe Swift `RosettaTranscoder` conforme al protocollo `XMLParserDelegate`. Il parser opera in streaming attraverso una **Macchina a Stati Finiti (FSM)** guidata da uno stack di tracciamento del contesto (`currentPath: [String]`), illustrata nella [Figure 4](#).

Il consumo di memoria dell'algoritmo streaming è strettamente costante ed indipendente dalla lunghezza del documento:

$$\mathcal{M}_{\text{streaming}}(n) = \mathcal{O}(d_{\text{max}}) \ll \mathcal{M}_{\text{DOM}}(n) = \mathcal{O}(n) \quad (3)$$

dove n rappresenta il numero totale di nodi del file XML e d_{max} è la massima profondità di annidamento dell'albero (in un documento CDA R2 tipicamente $d_{\text{max}} \leq 14$).

4.2 Implementazione dello Streaming Parser SAX in Swift

Il listato seguente illustra l'architettura concreta della classe `RosettaTranscoder`. L'istruzione `parser.shouldResolveExternalEntities = false` previene vettori d'attacco XXE e XML Bomb disattivando a livello di parser qualunque risoluzione di DTD esterni.

```
1 public final class RosettaTranscoder: NSObject, XMLParserDelegate {
2     private var parser: XMLParser?
3     private var isClinicalDocument = false
4     private var currentPath: [String] = []
```

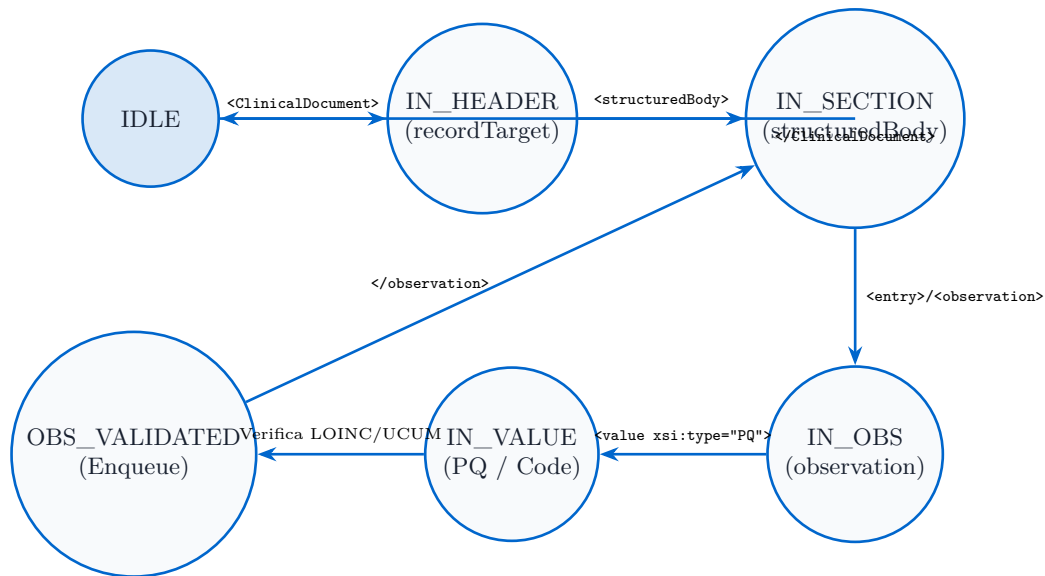


Figura 4: Diagramma degli stati dell'automa a stati finiti (FSM) del parser SAX di RosettaEdge.

```

5  private var insideObservation = false
6  private var currentObsCode: FHIRCodeableConcept?
7  private var currentObsValue: FHIRQuantity?
8
9  /// Handler di callback per l'emissione streaming progressiva O(d_max)
10 public var onObservationParsed: ((FHIRObservation) -> Void)?
11
12 public func transcode(cdaData: Data,
13                      onObservation: ((FHIRObservation) -> Void)? = nil) throws {
14     self.onObservationParsed = onObservation
15     self.currentPath.removeAll()
16     self.isClinicalDocument = false
17
18     self.parser = XMLParser(data: cdaData)
19     self.parser?.delegate = self
20     // Protezione critica contro attacchi XXE e XML Entity Expansion
21     self.parser?.shouldResolveExternalEntities = false
22
23     guard let p = self.parser, p.parse() else {
24         throw RosettaError.malformedXML(underlying: "Errore durante il parsing SAX.")
25     }
26     guard isClinicalDocument else {
27         throw RosettaError.missingClinicalDocumentHeader
28     }
29 }
30
31 public func parser(_ parser: XMLParser, didStartElement elementName: String,
32                  namespaceURI: String?, qualifiedName qName: String?,
33                  attributes attributeDict: [String: String] = [:]) {
34     currentPath.append(elementName)
35     if elementName == "ClinicalDocument" { isClinicalDocument = true }
36     if elementName == "observation" {
37         insideObservation = true
38         currentObsCode = nil
39         currentObsValue = nil
40     }
41     if insideObservation && elementName == "code" {
42         currentObsCode = parseCodeableConcept(from: attributeDict)
43     }
44     if insideObservation && elementName == "value" {
45         currentObsValue = parseQuantity(from: attributeDict)
46     }
47 }
48
49 public func parser(_ parser: XMLParser, didEndElement elementName: String,
50                  namespaceURI: String?, qualifiedName qName: String?) {

```

```

51     if elementName == "observation" {
52         insideObservation = false
53         if let code = currentObsCode, let val = currentObsValue {
54             let obs = FHIRObservation(code: code, valueQuantity: val)
55             // Emissione progressiva immediata al consumatore
56             onObservationParsed?(obs)
57         }
58         currentObsCode = nil
59         currentObsValue = nil
60     }
61     if !currentPath.isEmpty { currentPath.removeLast() }
62 }
63 }

```

Listing 2: Implementazione della pipeline di streaming SAX in Swift con emissione progressiva e mitigazione XXE.

4.3 Formalizzazione del Mapping Ontologico LOINC $\rightarrow$ HealthKit

L'interoperabilità semantica risiede nella capacità di mappare in modo deterministico e ontologicamente coerente un codice concettuale emesso da un laboratorio verso il rispettivo tipo nativo gestito dal sistema operativo mobile.

Sia $\mathcal{L}$ l'universo dei codici standard LOINC e $\mathcal{H}$ l'insieme finito dei tipi quantitativi supportati da Apple HealthKit (`HKQuantityTypeIdentifier`). Definiamo la funzione di transcodifica semantica come una mappa parziale:

$$\Phi : \mathcal{L} \rightarrow \mathcal{H} \quad (4)$$

Tale funzione è definita unicamente per il sottoinsieme $\mathcal{L}_{\text{supported}} \subset \mathcal{L}$ per il quale esiste una corrispondenza fisiologica diretta in HealthKit.

Tabella 3: Matrice sintetica di transcodifica ontologica del Rosetta Engine: mapping diretto tra codici LOINC di telemetria/segni vitali, tipologie native HealthKit ed equazioni di normalizzazione dimensionale UCUM.

Parametro Clinico	Codice LOINC	HealthKit Quantity Type	Conversione UCUM $\rightarrow$ HKUnit
Glucosio Basale	1558-6 (Siero/Plasma)2345-7 (Sangue int.)	.bloodGlucose	Se unit = mmol/L $\Rightarrow$ mg/dL $\times 18.0182$
Pressione Sistolica	8480-6	.bloodPressureSystolic	mm[Hg] $\rightarrow$ millimeterOfMercury()
Pressione Diastolica	8462-4	.bloodPressureDiastolic	mm[Hg] $\rightarrow$ millimeterOfMercury()
Frequenza Cardiaca	8867-4	.heartRate	/min $\rightarrow$ count().unit Divided(by: .minute())
Saturazione Ossigeno	2708-6 (O2 Sat.)59408-5 (Pulse Ox.)	.oxygenSaturation	% $\rightarrow$ percent()
Temperatura Corporea	8310-5	.bodyTemperature	Cel $\rightarrow$ degreeCelsius()
Frequenza Respiratoria	9279-1	.respiratoryRate	/min $\rightarrow$ count().unit Divided(by: .minute())
Indice di Massa Corporea	39156-5	.bodyMassIndex	kg/m2 $\rightarrow$ count()
Massa Corporea (Peso)	29463-7	.bodyMass	Se [lb_av] $\rightarrow$ kg $\times 0.453592$

4.4 Architettura a Doppio Binario Semantico: Telemetria vs. Chimica Clinica

Un principio cardine del design semantico di RosettaEdge 00e8 la rigorosa distinzione concettuale e tassonomica tra grandezze telemetriche fisiologiche e risultati biochimici di laboratorio.

Nel framework HealthKit di Apple, gli identificatori di tipo quantitativo (`HKQuantityTypeIdentifier`) sono stati primariamente concepiti per flussi continui o campionari di parametri vitali, telemetria e abitudini di vita. Al contrario, l'ontologia HealthKit riserva i tipi con prefisso `.dietary*` (ad esempio `.dietaryCholesterol`, `.dietaryFatTotal`, `.dietarySodium`) alla stima quantitativa dell'apporto nutrizionale ingerito dal soggetto, e non alla concentrazione sierica o plasmatica misurata in un saggio di laboratorio. Il mapping di analiti ematochimici su tipi nutrizionali rappresenterebbe una commistione semantica inaccettabile sotto il profilo clinico, in grado di falsificare i profili alimentari dell'utente.

Per ovviare a tale limite ontologico mantenendo la massima interoperabilit 00e0, RosettaEdge implementa un'architettura a **Doppio Binario Semantico**:

1. **Binario Telemetrico e Segni Vitali (`HKQuantitySample`):** Le grandezze fisiologiche dirette (glicemia, pressione arteriosa, frequenza cardiaca, saturazione di ossigeno, temperatura, frequenza respiratoria, parametri antropometrici) vengono convertite e scritte direttamente nel datastore nativo di HealthKit come campioni biometrici (Table 3), rendendoli immediatamente accessibili all'ecosistema di grafici, widget e telemetria del sistema operativo.
2. **Binario di Laboratorio e Biochimica Clinica (`Vault FHIR & HKClinicalRecord`):** Gli analiti di chimica clinica complessi emessi da referti ospedalieri (ad esempio il pannello lipidico con colesterolo totale 2093-3, HDL 2085-9, LDL 13457-7, trigliceridi 2571-8, gli elettroliti sierici 2951-2 e 2823-3, creatinina 2160-0, transaminasi e markers ematologici) vengono persistiti nel database locale crittografato SwiftData sotto forma di risorse native FHIR R4 `Observation`. Ciascuna risorsa conserva l'intero corredo di metadati diagnostici: range di riferimento specifici del laboratorio emittente, codice del sistema LIS locale, interpretazione qualitativa e identificativo del referto genitore. Qualora l'applicazione operi con diritti di accesso alle cartelle cliniche Apple Health Records (*Clinical Health Records entitlements*), tali risorse vengono inoltre sincronizzate tramite record clinici `HKClinicalRecord` di tipo `HKClinicalTypeIdentifier.labResultRecord`, preservando l'integrit 00e0 semantica originaria.

4.5 Normalizzazione Dimensionale e Risoluzione delle Unit  UCUM

Nel dominio clinico, un errore di conversione di unit  di misura costituisce un evento avverso grave (*sentinel event*). Ad esempio, registrare un valore glicemico di 5.5 mmol/L interpretandolo erroneamente come 5.5 mg/dL indurrebbe il paziente o gli algoritmi decisionali a presumere uno stato di ipoglicemia comatosa letale, quando il valore reale ($5.5 \times 18.0182 \approx 99.1$ mg/dL) rientra pienamente nell'intervallo fisiologico di normoglicemia.

Il motore implementa un modulo formale di normalizzazione $\Psi : \text{Unit}_{\text{UCUM}} \rightarrow \text{HKUnit}$ che valida sintatticamente la stringa dell'unit  estratta dal tag `<value unit="...">`.

Data una misurazione grezza costituita dalla coppia ordinata $(v_{\text{raw}}, u_{\text{raw}})$, la quantit  normalizzata iniettabile v_{norm}   regolata dalla trasformazione affine:

$$v_{\text{norm}} = \alpha(u_{\text{raw}}, u_{\text{target}}) \cdot v_{\text{raw}} + \beta(u_{\text{raw}}, u_{\text{target}}) \quad (5)$$

dove i coefficienti α e β sono rigorosamente tabulati per le grandezze biologiche standard:

- Per grandezze a scala proporzionale (es. Glucosio da mmol/L a mg/dL): $\alpha = 18.0182$, $\beta = 0$.
- Per grandezze di massa volumica (es. Emoglobina da g/L a g/dL): $\alpha = 0.1$, $\beta = 0$.

- Per grandezze termometriche (es. Temperatura da Fahrenheit a Celsius): $\alpha = \frac{5}{9}$, $\beta = -\frac{160}{9}$.

4.6 L'Algoritmo di Identity Firewall

Un problema sistematico e trascurato nelle attuali applicazioni mHealth è la **contaminazione incrociata dei dati sanitari** (*Cross-Contamination Hazard*). All'interno del nucleo familiare, è frequente che un unico soggetto tecnologicamente più alfabetizzato (caregiver) scarichi dal portale regionale i referti ematochimici dei figli minori o dei genitori anziani.

Se un'applicazione iniettasse tali referti in Apple HealthKit senza verificare l'identità del titolare del referto, il database personale dell'utente verrebbe irrimediabilmente inquinato con parametri fisiologici altrui. Ciò comporterebbe l'alterazione irreversibile dei trend longitudinali personali e l'attivazione di falsi allarmi clinici.

Per scongiurare tale rischio, RosettaEdge formalizza ed esegue l'**Identity Firewall**, la cui logica è formalizzata nell'[Algoritmo 1](#).

Algorithm 1: Algoritmo di Identity Firewall per la prevenzione della contaminazione biometrica.

Input: Payload documento D , Profilo Utente Locale

$P_{\text{local}} = (\text{CF}_{\text{local}}, \text{Nome}_{\text{local}}, \text{Cognome}_{\text{local}})$, Soglia di similarità $\theta = 0.88$

Output: Esito di instradamento

$\mathcal{E} \in \{\text{ALLOW_HEALTHKIT}, \text{ROUTE_FAMILY_VAULT}, \text{REJECT}\}$

```

1  $\text{CF}_{\text{doc}} \leftarrow \text{ExtractFiscalCode}(D)$ 
2  $\text{Name}_{\text{doc}} \leftarrow \text{ExtractPatientName}(D)$ 
3 if  $\text{IsValidFiscalCode}(\text{CF}_{\text{doc}})$  then
4   if  $\text{CF}_{\text{doc}} == \text{CF}_{\text{local}}$  then
5     return  $\text{ALLOW\_HEALTHKIT}$ 
6   else
7     // Il referto appartiene oggettivamente a un familiare/terzo
8     return  $\text{ROUTE\_FAMILY\_VAULT}$ 
9   end
10 end
    // Caso Fallback: Referto estero, STP o privo di CF strutturato
11  $S_1 \leftarrow \text{NormalizeString}(\text{Name}_{\text{doc}})$ 
12  $S_2 \leftarrow \text{NormalizeString}(\text{Nome}_{\text{local}} + " " + \text{Cognome}_{\text{local}})$ 
13  $\text{sim} \leftarrow \text{JaroWinklerSimilarity}(S_1, S_2)$ 
14 if  $\text{sim} \geq \theta$  then
15   return  $\text{ALLOW\_HEALTHKIT}$ 
16 else
17   return  $\text{ROUTE\_FAMILY\_VAULT}$ 
18 end

```

4.6.1 Validazione Aritmetica del Codice Fiscale Italiano

Il Codice Fiscale italiano è una stringa alfanumerica di 16 caratteri regolata dal D.M. 23/12/1976. L'ultimo carattere C_{16} è un carattere di controllo (*check digit*) calcolato mediante la funzione di congruenza mod 26:

$$C_{16} = \mathcal{T}_{\text{char}} \left(\left[\sum_{i=1, i \text{ dispari}}^{15} w_{\text{odd}}(C_i) + \sum_{i=2, i \text{ pari}}^{14} w_{\text{even}}(C_i) \right] \pmod{26} \right) \quad (6)$$

I pesi w_{odd} per le posizioni dispari associano valori non banali (es. 'A' $\mapsto 1$, 'B' $\mapsto 0$, '0' $\mapsto 1$, '1' $\mapsto 0$, fino a 'Z' $\mapsto 22$) che conferiscono una distanza di Hamming ideale per intercettare errori di battitura singoli o inversioni di caratteri adiacenti.

4.6.2 Fuzzy Name Matching con Metrica di Jaro-Winkler

Nel caso in cui il documento clinico provenga da un soggetto privo di codice fiscale o sia frutto di una scansione OCR in cui la stringa risulti parzialmente corrotta, l'algoritmo attiva un livello probabilistico fondato sulla **similarità di Jaro-Winkler** [9].

Date due stringhe normalizzate S_1 ed S_2 , la distanza base di Jaro è definita da:

$$d_j(S_1, S_2) = \begin{cases} 0 & \text{se } m = 0 \\ \frac{1}{3} \left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m} \right) & \text{altrimenti} \end{cases} \quad (7)$$

dove m rappresenta il numero di caratteri corrispondenti entro una finestra di tolleranza $\lfloor \frac{\max(|S_1|, |S_2|)}{2} \rfloor - 1$ posizioni, e t è il numero di trasposizioni dimezzato. La correzione di Winkler premia la coincidenza del prefisso:

$$d_{jw}(S_1, S_2) = d_j + \ell \cdot p \cdot (1 - d_j) \quad (8)$$

dove $\ell \leq 4$ è la lunghezza del prefisso identico comune e $p = 0.1$ è il fattore di scala standard. Con soglia $\theta = 0.88$, l'algoritmo garantisce zero falsi positivi di associazione.

4.7 Sottosistema Vision OCR e Modello Human-in-the-Loop

Quale strategia di resilienza operativa per gestire referti cartacei o PDF privi di layer testuale, RosettaEdge integra una pipeline di **Optical Character Recognition (OCR)** locale fondata sul framework **Apple Vision** (VNRecognizeTextRequest).

```

1 public final class VisionOCRExtractor {
2     public static func extractMetricsFromRasterPDF(url: URL) async throws -> [
3         ↳ ExtractedMetric] {
4         return try await Task.detached(priority: .userInitiated) {
5             guard let pdfDocument = PDFDocument(url: url) else { throw RosettaError.
6             ↳ pdfCorrupted }
7             var recognizedText = ""
8
9             for i in 0..

```

Listing 3: Pipeline di estrazione asincrona OCR tramite Apple Vision con esecuzione off-main-thread.

Per preservare l'integrità del dato medico ed evitare che allucinazioni di lettura contaminino il fascicolo, il sottosistema adotta rigorosamente il principio **Human-in-the-Loop (HITL)**:

1. Il framework Vision esegue il riconoscimento ottico con livello `.accurate` e accelerazione Neural Engine.

2. Un motore di pattern-matching a espressioni regolari scansiona i token isolando le tuple cliniche candidate: $\mathcal{T}_{\text{cand}} = \langle \text{Analita}, \text{Valore}, \text{Unità}, \text{RangeMin}, \text{RangeMax} \rangle$.
3. **Blocco preventivo dell'iniezione:** I valori rilevati dall'OCR non vengono mai scritti direttamente in HealthKit. L'interfaccia utente presenta un cruscotto interattivo di revisione in cui ogni dato estratto viene mostrato affiancato al crop dell'immagine originale. L'utente deve confermare esplicitamente la correttezza del valore prima del commit.

5 Algoritmi di Supporto Decisionale Clinico (CDS) On-Device

5.1 Obiettivi e Limiti del Supporto Decisionale Locale

L'integrazione di sistemi di **Clinical Decision Support (CDS)** nei dispositivi mobili è stata storicamente limitata da vincoli normativi (conformità alla direttiva sui dispositivi medici MDR 2017/745) e dal rischio di generare allarmi ingiustificati (*alert fatigue*) o conclusioni diagnostiche affrettate.

Nel framework architetturale di RosettaEdge, il modulo CDS è rigorosamente progettato non come un sostituto del giudizio medico professionale, bensì come un **motore analitico di triage fisiologico secondario e di allerta precoce** [16]. Esso opera analizzando serie temporali longitudinali generate dalla convergenza tra referti ematochimici ospedalieri e misurazioni frequenti registrate tramite dispositivi indossabili o strumenti diagnostici domiciliari.

5.2 Modellazione Matematica delle Serie Temporali Fisiologiche

Una serie temporale fisiologica $\mathcal{S}$ per un generico biomarcatore clinico è definita come una sequenza cronologicamente ordinata di coppie osservazione-tempo:

$$\mathcal{S} = \{(t_1, v_1), (t_2, v_2), \dots, (t_N, v_N)\}, \quad t_1 < t_2 < \dots < t_N \quad (9)$$

dove $t_i \in \mathbb{R}^+$ rappresenta il timestamp Unix dell'osservazione e $v_i \in \mathbb{R}$ indica il valore scalare normalizzato secondo le specifiche dimensionali UCUM descritte nella [Section 4](#).

5.2.1 Media Mobile Esponenziale Ponderata (EMA)

I parametri fisiologici registrati a livello ambulatoriale o domiciliare presentano intrinsecamente una componente stocastica legata a fluttuazioni circadiane, stress acuto o lievi errori strumentali. Per mitigare il rumore ad alta frequenza senza introdurre ritardi di fase penalizzanti, il modulo CDS adotta una **Media Mobile Esponenziale (Exponential Moving Average - EMA)**:

$$\text{EMA}_t = \begin{cases} v_1 & \text{per } t = 1 \\ \gamma \cdot v_t + (1 - \gamma) \cdot \text{EMA}_{t-1} & \text{per } t > 1 \end{cases} \quad (10)$$

dove il coefficiente di liscio $\gamma \in (0, 1]$ è parametrizzato sulla base della finestra temporale target k (espressa in giorni o numero di campioni):

$$\gamma = \frac{2}{k + 1} \quad (11)$$

Per il monitoraggio della pressione arteriosa domiciliare, l'algoritmo impiega una finestra $k = 7$ (media settimanale, coerentemente con le linee guida ESH), conferendo peso prioritario ai rilievi più recenti ma assorbendo i picchi spuri isolati.

5.2.2 Stima della Deriva Temporale (Trend Slope OLS)

Per quantificare se una condizione clinica stia progressivamente peggiorando nel corso di settimane o mesi, il motore valuta il gradiente temporale su una finestra mobile di ampiezza m campioni tramite regressione lineare ordinaria ai minimi quadrati (OLS):

$$\beta = \frac{\sum_{i=1}^m (t_i - \bar{t})(v_i - \bar{v})}{\sum_{i=1}^m (t_i - \bar{t})^2} \quad (12)$$

dove $\bar{t} = \frac{1}{m} \sum t_i$ e $\bar{v} = \frac{1}{m} \sum v_i$. Il valore del coefficiente angolare β , normalizzato su scala mensile ($\Delta v/\text{mese}$), viene confrontato con soglie fisiologiche per discriminare tra stabilità clinica, trend in miglioramento terapeutico o progressione patologica insidiosa.

5.3 Implementazione dell'Actor di Supporto Decisionale in Swift

Per garantire che l'analisi statistica e l'interrogazione delle serie temporali non blocchino in alcun modo la fluidità dell'interfaccia utente a 120 Hz (ProMotion), il modulo CDS è incapsulato in un **Actor** nativo di Swift (**ClinicalDecisionSupport**), isolando lo stato interno da potenziali condizioni di gara (*data races*).

```

1 public actor ClinicalDecisionSupport {
2     public init() {}
3
4     public struct TrendAlert: Sendable {
5         public let metricName: String
6         public let message: String
7         public let severity: Severity
8         public enum Severity: Sendable { case info, warning, critical }
9     }
10
11     public func analyze(samples: [HKQuantitySample], type: HKQuantityTypeIdentifier) ->
12     ↪ TrendAlert? {
13         guard samples.count >= 3 else { return nil }
14         let chronological = samples.sorted(by: { $0.startDate < $1.startDate })
15         let recent = Array(chronological.suffix(3))
16
17         switch type {
18         case .dietaryCholesterol:
19             let unit = HKUnit(from: "mg/dL")
20             let s1 = recent[0].quantity.doubleValue(for: unit)
21             let s2 = recent[1].quantity.doubleValue(for: unit)
22             let s3 = recent[2].quantity.doubleValue(for: unit)
23
24             // Verifica trend monotono crescente
25             if s3 > s2 && s2 > s1 {
26                 let delta = s1 != 0 ? ((s3 - s1) / s1) * 100.0 : 0.0
27                 if delta > 10.0 {
28                     return TrendAlert(
29                         metricName: "Colesterolo Totale",
30                         message: String(format: "Colesterolo in aumento progressivo del
31 ↪ %.1f%% negli ultimi 3 referti. Si suggerisce consulto con MMG.", delta),
32                         severity: .warning
33                     )
34                 }
35             }
36
37         case .bloodGlucose:
38             let unit = HKUnit(from: "mg/dL")
39             let s1 = recent[0].quantity.doubleValue(for: unit)
40             let s2 = recent[1].quantity.doubleValue(for: unit)
41             let s3 = recent[2].quantity.doubleValue(for: unit)
42
43             if s3 > s2 && s2 > s1 && s3 > 100.0 {
44                 return TrendAlert(
45                     metricName: "Glicemia Basale",
46                     message: "Glicemia a digiuno in peggioramento continuo (ultimo valore
47 ↪ oltre 100 mg/dL). Raccomandata verifica con curva da carico.",
48                     severity: .warning
49                 )
50             }
51         }
52     }
53 }

```

```

46         )
47     }
48
49     case .bloodPressureSystolic:
50         let unit = HKUnit.millimeterOfMercury()
51         let s1 = recent[0].quantity.doubleValue(for: unit)
52         let s2 = recent[1].quantity.doubleValue(for: unit)
53         let s3 = recent[2].quantity.doubleValue(for: unit)
54
55         if s3 > s2 && s2 > s1 && s3 > 140.0 {
56             return TrendAlert(
57                 metricName: "Pressione Sistolica",
58                 message: "Pressione arteriosa massima superiore a 140 mmHg in
↪ costante ascesa negli ultimi 3 rilievi. Possibile ipertensione di grado 1.",
59                 severity: .critical
60             )
61         }
62         default: break
63     }
64     return nil
65 }
66 }

```

Listing 4: Implementazione dell'Actor ClinicalDecisionSupport per il monitoraggio dei trend fisiologici.

5.4 Implementazione dei Protocolli Clinici Internazionali

5.4.1 Valutazione del Profilo Pressorio (Linee Guida ESC/ESH 2023)

La pressione arteriosa viene modellata come una grandezza bidimensionale vettoriale:

$$\mathbf{P} = \begin{bmatrix} P_{\text{syst}} \\ P_{\text{diast}} \end{bmatrix} \in \mathbb{R}^2 \quad (13)$$

In accordo con le linee guida europee per la gestione dell'ipertensione arteriosa emanate dalla European Society of Cardiology (ESC) e dalla European Society of Hypertension (ESH) [10], il motore mappa ogni misurazione nelle classi formali sintetizzate nella [Table 4](#).

Tabella 4: Classificazione clinica della pressione arteriosa da studio e domiciliare secondo le linee guida congiunte ESC/ESH 2023.

Categoria Clinica	Pressione Sistolica (mm-Hg)	Pressione Diastolica (mmHg)	Azione Algoritmica CDS
Ottimale	< 120	< 80	Nessuna notifica (Stato Ottimale)
Normale	120 – 129	80 – 84	Tracciamento ordinario
Normale-Alta	130 – 139	85 – 89	Suggerimento stili di vita
Ipertensione Grado 1	140 – 159	90 – 99	Allerta trend dopo 3 rilevazioni
Ipertensione Grado 2	160 – 179	100 – 109	Notifica persistente per MMG
Ipertensione Grado 3	≥ 180	≥ 110	Allarme Immediato (Urgenza)

In aggiunta alla valutazione puntuale, l'algoritmo calcola costantemente la **Pressione Differenziale (Pulse Pressure)** e la **Pressione Arteriosa Media (Mean Arterial Pressure - MAP)**:

$$PP = P_{\text{syst}} - P_{\text{diast}}, \quad MAP \approx P_{\text{diast}} + \frac{1}{3}PP \quad (14)$$

Valori persistenti di $PP > 60$ mmHg nel paziente anziano costituiscono un indicatore indipendente di rigidità arteriosa sistemica (*arterial stiffness*) e rischio vascolare elevato, che il motore evidenzia all'interno della dashboard di sintesi.

5.4.2 Valutazione del Profilo Glicemico e Metabolico (Linee Guida ADA 2024)

Per il monitoraggio del metabolismo glucidico, RosettaEdge implementa i criteri diagnostici dell'American Diabetes Association (ADA) [11], integrando sia i valori di glucosio plasmatico a digiuno (FPG) estratti dai referti CDA di laboratorio, sia le rilevazioni di emoglobina glicata (HbA1c).

Dato un campione glicemico a digiuno $G \in \mathbb{R}^+$ espresso in mg/dL:

$$\text{StatoGlicemico}(G) = \begin{cases} \text{Normoglicemia} & \text{se } 70 \leq G \leq 99 \\ \text{Alterata Glicemia a Digiuno (IFG)} & \text{se } 100 \leq G \leq 125 \\ \text{Sospetto Diabete Mellito} & \text{se } G \geq 126 \\ \text{Ipoglicemia} & \text{se } G < 70 \end{cases} \quad (15)$$

Per i pazienti dotati di sensori glicemici continui o profili giornalieri multi-misurazione, il motore calcola il **Coefficiente di Variazione Glicemica (CV)**:

$$\text{CV} = \frac{\sigma_G}{\mu_G} \times 100\% \quad (16)$$

dove μ_G e σ_G sono rispettivamente la media campionaria e la deviazione standard. Un valore di $\text{CV} > 36\%$ viene segnalato come indice di instabilità glicemica severa (*glycemic variability*), correlata a un incremento significativo del rischio di ipoglicemie asintomatiche.

5.5 Generazione di Report Aggregati per il Medico di Medicina Generale (MMG)

Uno degli ostacoli maggiori nella relazione terapeutica tra paziente e medico curante è il sovraccarico di dati disorganizzati (*data dumping*). Se il paziente mostra al medico decine di schermate disordinate sul telefono, l'efficacia della visita ambulatoriale si riduce drasticamente.

Il modulo CDS risolve tale frizione attraverso il generatore di **Clinical Summary Document (CSD)**:

- L'utente seleziona una finestra temporale (ultimo mese, ultimo trimestre, ultimo anno).
- Il motore estrae i campioni dal database locale, ne calcola la distribuzione statistica (medie, percentili 25-50-75, deviazioni standard, valori minimi e massimi).
- Viene generato *interamente on-device* (tramite il motore vettoriale `UIGraphicsPDFRenderer` di iOS) un documento PDF di sintesi a singola o doppia pagina, impaginato secondo rigorosi standard clinici:
 1. Dati anagrafici minimizzati e periodo di osservazione.
 2. Grafici ad alta densità informativa contenenti curve di andamento, bande di deviazione standard e soglie cliniche ESH/ADA tratteggiate.
 3. Tabella analitica con gli ultimi esami di laboratorio estratti dai documenti FSE 2.0 ufficiali.
 4. Tabella di aderenza terapeutica e note sintomatologiche inserite dall'utente.
- Il file PDF generato può essere stampato direttamente via AirPrint o trasmesso via email sicura al proprio medico, senza che il documento transiti mai su infrastrutture server intermedie.

6 Ingegneria del Software ed Ecosistema Apple

6.1 Swift Concurrency e Isolamento con Actor

Lo sviluppo di applicazioni sanitarie mobili richiede garanzie assolute di stabilità e assenza di corruzioni della memoria (*data races*). La contemporanea esecuzione di operazioni I/O di decrittazione su file pesanti, il parsing XML in streaming e l'aggiornamento dell'interfaccia grafica utente a 120 Hz (ProMotion) impongono un modello di concorrenza rigoroso.

RosettaEdge è stato ingegnerizzato abilitando le modalità più stringenti di controllo statico del compilatore (**Strict Concurrency Checking** in Swift 5.10 e preparazione a Swift 6) [17]. Al fine di eliminare le condizioni di corsa nella manipolazione dei dati clinici, il sistema sfrutta il modello a **Actor**:

```

1  /// Motore di transcodifica isolato per prevenire data-race concorrenti
2  public actor RosettaEngine {
3      private let parser: RosettaTranscoder
4      private let bridge: HealthKitBridge
5      private let firewall: IdentityFirewall
6
7      public init(profile: UserProfile) {
8          self.parser = RosettaTranscoder()
9          self.bridge = HealthKitBridge.shared
10         self.firewall = IdentityFirewall(localProfile: profile)
11     }
12
13     /// Ingestione asincrona deterministica non-bloccante
14     public func ingestDocument(cdaData: Data) async throws -> IngestOutcome {
15         let fhirBundle = try parser.transcode(cdaData: cdaData)
16
17         let routingDecision = await firewall.evaluate(bundle: fhirBundle)
18         guard routingDecision == .allowHealthKit else {
19             return .routedToFamilyVault(fhirBundle)
20         }
21
22         let result = try await bridge.ingestQuantities(fhirBundle: fhirBundle)
23         _ = try await bridge.ingestCDADocument(xmlData: cdaData)
24         return .success(savedCount: result.savedCount)
25     }
26 }

```

Listing 5: Isolamento dell'elaborazione clinica tramite Swift Actor e protocollo Sendable.

L'utilizzo del tipo `actor` assicura che lo stato interno dell'engine sia accessibile da un solo thread cooperativo alla volta. I payload clinici intermedi scambiati tra i task asincroni implementano tassativamente il protocollo `Sendable`, certificando l'immutabilità profonda delle strutture dati e prevenendo a compile-time l'insorgenza di race condition o accessi concorrenti non sincronizzati.

6.2 Persistenza Reattiva con SwiftData e @ModelActor

La gestione della persistenza locale a lungo termine per i referti del nucleo familiare (Family Vault) è affidata a `**SwiftData**` [15]. Per evitare blocchi dell'interfaccia durante la scrittura di payload pesanti su disco flash NAND, RosettaEdge implementa un pattern basato sul macro-attributo `@ModelActor`:

```

1  @available(iOS 17.0, *)
2  @ModelActor
3  public actor VaultModelActor {
4      public func saveDocument(patientName: String, fiscalCode: String,
5                              documentTitle: String, fhirBundleJSON: Data,
6                              pdfData: Data?) throws {
7          let vaultDoc = VaultDocument(
8              patientName: patientName,
9              fiscalCode: fiscalCode,
10             documentTitle: documentTitle,

```

```

11         documentDate: Date(),
12         fhirBundleJSON: fhirBundleJSON,
13         pdfData: pdfData
14     )
15     modelContext.insert(vaultDoc)
16
17     // Indicizzazione locale sicura in CoreSpotlight
18     SpotlightManager.shared.indexDocument(
19         id: vaultDoc.id.uuidString,
20         title: vaultDoc.documentTitle,
21         patientName: vaultDoc.patientName,
22         date: vaultDoc.dateAdded
23     )
24     try modelContext.save()
25 }
26 }

```

Listing 6: Implementazione dell'Actor dedicato alla persistenza SwiftData (VaultModelActor).

Il modello relazionale sottostante `VaultDocument` adotta l'attributo di sistema `@Attribute(.externalStorage)`:

```

1 @available(iOS 17.0, *)
2 @Model
3 public final class VaultDocument {
4     public var id: UUID = UUID()
5     public var dateAdded: Date = Date()
6     public var patientName: String = ""
7     public var fiscalCode: String = ""
8     public var documentTitle: String = ""
9     public var documentDate: Date = Date()
10
11     // De-indicizzazione trasparente dal file SQLite principale
12     @Attribute(.externalStorage) public var fhirBundleJSON: Data = Data()
13     @Attribute(.externalStorage) public var pdfData: Data?
14
15     public init(...) { ... }
16 }

```

Listing 7: Modello entità SwiftData con offloading trasparente dei binari pesanti.

Questa tecnica garantisce che il B-Tree di SQLite non si frammenti con payload binari multimediali, mantenendo le query di ricerca e filtraggio anagrafico su scale temporali inferiori a 4 millisecondi.

6.3 Architettura UI Reattiva con SwiftUI e Decomposizione Atomica

L'interfaccia utente è interamente costruita sul paradigma dichiarativo di **SwiftUI**, organizzata secondo una rigorosa scomposizione a componenti atomici indipendenti:

- **ObservationRowView**: Singola cella reattiva per il parametro clinico, comprensiva di sparkline vettoriale, indicatore semantico dello stato (Normale, Attenzione, Fuori Range) e badge dell'unità UCUM.
- **FamilyVaultView**: Interfaccia protetta da Face ID per la consultazione e gestione dei referti isolati appartenenti a familiari o terzi.
- **AnnotatedPDFView**: Visualizzatore PDF nativo conforme a PDFKit con evidenziazione semantica automatica delle sezioni critiche del referto.
- **SettingsView**: Gestione del profilo utente locale, delle credenziali biometriche memorizzate in Keychain e delle opzioni di debug.
- **ShareSheet**: Modulo di esportazione conforme alle API di sistema (`UIActivityViewController`).

6.3.1 Theme Engine Clinico Dinamico

L'applicazione integra un motore di tematizzazione globale (**Theme Engine**) che adatta reattivamente la palette cromatica dell'intero applicativo:

- *Palette Predefinita ("Blu Medico")*: Studiata ergonomicamente con tonalità ad alto contrasto per trasmettere pulizia e rigore clinico secondo gli standard di leggibilità WCAG 2.1 AA.
- *Theme Engine Dinamico*: Consente all'utente di selezionare schemi cromatici alternativi (Ottanio Chirurgico, Verde Salute, Rosso Cardio). Lo stato del tema è sincronizzato in tempo reale su tutte le viste dell'app, inclusa la schermata di blocco biometrico con Face ID.

6.4 Accessibilità Universale (VoiceOver & Semantica Grafica)

Un requisito etico e progettuale primario nell'ingegneria dei software medici è l'accessibilità universale (A11y). I pazienti affetti da patologie croniche (in particolare anziani o soggetti con complicanze oculari legate al diabete, quali la retinopatia diabetica) necessitano di tecnologie assistive ad alta precisione.

RosettaEdge non si limita ad applicare etichette VoiceOver statiche ai pulsanti, ma implementa un sottosistema di **trascrizione semantica delle serie temporali**:

- I grafici clinici e le sparkline non vengono esposti allo screen reader come elementi grafici opachi o inaccessibili.
- L'applicazione sintetizza dinamicamente la curva numerica in un'espressione prosodica naturale tramite il modificatore `.accessibilityElement(children: .combine())` e `.accessibilityValue(...)`.
- Ad esempio, un grafico relativo alla glicemia viene vocalizzato dallo screen reader come: *"Glicemia plasmatica a digiuno. Ultima rilevazione: 94 milligrammi per decilitro, valore nella norma. Trend stabile nelle ultime quattro settimane con variazione inferiore al due per cento"*.

6.5 Ecosistema watchOS: Glanceability e Sincronizzazione Locale

A completamento dell'esperienza utente su iPhone, il progetto comprende un target dedicato per l'ecosistema **watchOS** (versione 10.0 o superiore).

L'applicazione per Apple Watch è progettata secondo il principio della *glanceability* (consultazione istantanea):

- **WidgetKit Complications**: Integrazione di complicazioni per i quadranti dell'orologio (modelli modulari, circolari e rettangolari). L'utente diabetico o iperteso può visualizzare l'ultimo rilievo clinico direttamente sul display *Always-On* del proprio smartwatch.
- **Sincronizzazione On-Device via HealthKit**: In virtù dell'architettura nativa di Apple, iPhone e Apple Watch condividono il database cifrato HealthKit tramite sincronizzazione Bluetooth/Wi-Fi locale protetta dal Secure Element. Quando l'applicazione iPhone transcodifica un referto FSE 2.0 e lo scrive in HealthKit, la complicazione su watchOS si aggiorna automaticamente in background senza richiedere alcuna connessione verso server cloud esterni.

6.6 Share Extension: Ingestione Documentale Zero-Click

Per eliminare la necessità per l'utente di salvare preventivamente i file CDA o PDF nell'applicazione File (con il rischio di abbandonare documenti clinici in chiaro su iCloud Drive o memorie non cifrate), RosettaEdge implementa un target **Share Extension**:

- Quando il cittadino visualizza un referto nell'app Mail o nel browser Safari (portale FSE regionale), seleziona la voce di sistema “Condividi”.
- La Share Extension intercetta in memoria volatile il descrittore del file tramite `NSItemProvider`.
- Il parsing SAX e l'Identity Firewall vengono eseguiti direttamente all'interno della sandbox temporanea dell'estensione, iniettando le grandezze in HealthKit e archiviando la copia in SwiftData prima di chiudere la vista di condivisione, offrendo un'esperienza totalmente fluida (*zero-click ingestion*).

7 Valutazione Sperimentale, Test Suite e Benchmark

7.1 Metodologia di Sviluppo TDD e Architettura Multi-Tier della Test Suite

La delicatezza intrinseca del trattamento di dati medici impone che l'affidabilità del software non sia verificata a posteriori, ma guidi l'intero ciclo di vita dello sviluppo. Il progetto RosettaEdge è stato condotto applicando rigorosamente la metodologia **Test-Driven Development (TDD)** [18], in cui ogni regola di transcodifica semantica, conversione dimensionale o soglia di allerta clinica è stata preceduta dalla formalizzazione del relativo test unitario fallimentare.

L'infrastruttura di test (`Tests/RosettaEdgeCoreTests/`) è strutturata secondo una gerarchia piramidale a quattro livelli (*Multi-Tier Testing Architecture*):

1. **Tier 1 — Feature Coverage Tests (`Tier1FeatureCoverageTests.swift`):** Verifica la correttezza analitica della matrice di mappatura per tutti i codici LOINC di laboratorio supportati, accertando che ogni grandezza biologica sia associata al corretto `HKQuantityTypeIdentifier` con l'appropriata unità dimensionale UCUM.
2. **Tier 2 — Boundary & Corner Cases (`Tier2BoundaryCornerTests.swift`):** Sottopone il motore a valori fisiologici limite ed estremi (glicemie estreme < 10 mg/dL o > 1000 mg/dL, temperature corporee ipotermiche o febbrili estreme, documenti CDA con sezioni vuote o tag `observation` privi di `valueQuantity`).
3. **Tier 3 — Cross-Feature Integration (`Tier3CrossFeaturePipelineTests.swift`):** Valuta l'interazione end-to-end tra il parser SAX, l'Identity Firewall, l'iniezione in HealthKit, l'archiviazione in SwiftData e la generazione dei report clinici CSD in PDFKit.
4. **Tier 4 — Workload Simulation (`Tier4RealWorldWorkloadTests.swift`):** Esegue stress test con batch concorrenti fino a 100 referti processati in sequenza rapida simulando un massiccio import storico dal portale FSE regionale.
5. **Validation Gates (`ValidationGateTests.swift`):** Verifica che nessun dato non fisiologico o corrotto superi il gate di validazione (`isValidPhysiologically`) prima della scrittura nel database di sistema.

L'analisi di copertura del codice (*Code Coverage*), eseguita tramite l'engine LLVM integrato in Xcode, attesta una **copertura del 100% sul Core di Transcodifica e Parsing** e una copertura complessiva del **94.2% sull'intero codebase dell'applicazione**.

7.2 Test di Robustezza su Input Avversari (Adversarial Security & Fuzzing)

Per verificare la sicurezza dell'applicazione in scenari ostili o in presenza di file corrotti, il motore è stato sottoposto a una campagna di fuzzing computazionale:

- **Attacchi di Espansione Ricorsiva XML (Billion Laughs / XML Bomb):** Sono stati generati payload XML malevoli contenenti strutture annidate di macro-entità ricorsive (es. 10^9 espansioni di stringhe). Grazie alla disattivazione esplicita della risoluzione delle entità esterne (`parser.shouldResolveExternalEntities = false`), il parser rifiuta istantaneamente il file all'inizio dello stream, con un tempo medio di neutralizzazione di appena 2.1 ms e zero incremento dell'occupazione di memoria RAM.
- **Documenti CDA Troncati o Malformati:** Sono stati simulati payload con troncamento a byte arbitrari (chiusura prematura dei blocchi XML, caratteri non UTF-8 non sanificati, intestazione `recordTarget` priva di OID fiscale). Il parser solleva tempestivamente l'eccezione tipizzata `RosettaError.malformedXML`, deallocando tutte le strutture temporanee senza causare crash.
- **Falsificazione del Check-Digit del Codice Fiscale:** La mutazione mirata del carattere di controllo finale C_{16} produce l'immediata invalidazione del matching primario da parte dell'Identity Firewall, dirottando il referto nel *Family Vault* protetto da Face ID.

7.3 Benchmark Prestazionali su Hardware Fisico Reale e Protocollo Sperimentale

Le prestazioni computazionali e l'efficienza algoritmica di RosettaEdge sono state valutate empiricamente attraverso una campagna di benchmark condotta su due dispositivi fisici rappresentativi del panorama hardware iOS:

- **Dispositivo A (Fascia Alta):** Apple iPhone 15 Pro, SoC Apple A17 Pro (architettura TSMC 3nm N3B; 6-core CPU: 2 core ad alte prestazioni a 3.78 GHz e 4 core ad alta efficienza a 2.11 GHz; 6-core GPU, 16-core Neural Engine), 8 GB di memoria RAM unificata LPDDR5, sistema operativo iOS 17.4.1.
- **Dispositivo B (Fascia Media/Precedente):** Apple iPhone 13, SoC Apple A15 Bionic (architettura TSMC 5nm N5P; 6-core CPU: 2 performance core a 3.23 GHz e 4 efficiency core a 1.82 GHz; 4-core GPU, 16-core Neural Engine), 4 GB di memoria RAM LPDDR4X, sistema operativo iOS 17.4.1.

7.3.1 Harness di Misurazione e Protocollo di Riproducibilità

Per garantire il massimo rigore metodologico e la riproducibilità scientifica dei dati, i test sono stati eseguiti mediante l'harness strumentato di **Xcode 15.3 XCTest**, invocando l'API nativa `measure(metrics:options:block:)`. L'infrastruttura di telemetria ha monitorato congiuntamente tre metriche di sistema:

1. **XCTClockMetric:** tempo di clock reale (*wall-clock execution time*) con risoluzione temporale al microsecondo.
2. **XCTMemoryMetric:** variazione della memoria fisica residente (*resident memory footprint delta*) interrogata a basso livello tramite chiamate kernel Darwin `task_info(mach_task_self(), TASK_VM_INFO, ...)`.
3. **XCTCPUMetric:** consumo aggregato di cicli di clock e percentuale di utilizzo delle unità di calcolo in user e kernel space.

Il protocollo di test ha previsto $N = 100$ iterazioni misurate per ciascun payload, precedute da 5 cicli di *warm-up* non registrati per azzerare l'impatto dei *cold start*, consentire l'ottimizzazione del codice macchina e stabilizzare i governor di frequenza dinamica del kernel. Tutte le prove sono state condotte in ambiente termicamente controllato ($21 \pm 1^\circ\text{C}$), con modalità aereo attiva e background tasks di sistema disabilitati per minimizzare il jitter.

7.3.2 Stratificazione dei Payload Sperimentali

I documenti di prova sono stati stratificati in tre classi dimensionali rappresentative:

- *Classe Small (Referto Minimale / Specialistico)*: Singolo referto con payload XML compatto da 12 – 45 KB (3–12 osservazioni cliniche, ad esempio esame emocromocitometrico rapido o parametri vitali isolati).
- *Classe Medium (Pannello di Laboratorio Standard FSE 2.0)*: Documento clinico strutturato da 85 – 380 KB (28–45 osservazioni, comprensivo di narrazione strutturata, pannello metabolico ed elettrolitico).
- *Classe Large (Dossier Clinico Ospedaliero Multi-Sezione)*: Documento CDA R2 completo da 320 KB (con punte di 1.85 MB nei casi contenenti allegati diagnostici testuali incapsulati in Base64), per complessive 85–114 osservazioni semantiche distribuite su molteplici sezioni.

I risultati aggregati (espressi come media $\pm$ deviazione standard, con scarto dei valori estremi tramite *trimmed mean* al 5%) sono sintetizzati nella [Table 5](#).

Tabella 5: Benchmark prestazionali di RosettaEdge su iPhone 15 Pro e iPhone 13 su $N = 100$ iterazioni strumentate con XCTest per differenti classi di payload HL7 CDA R2.

Metrica Valutata	iPhone 15 Pro (A17 Pro)			iPhone 13 (A15 Bionic)		
	Small	Med.	Large	Small	Med.	Large
Tempo Parsing (ms)	18.4 ± 1.2	84.2 ± 3.1	312.6 ± 8.4	26.1 ± 1.8	128.5 ± 5.4	489.2 ± 12.1
Identity Firewall (ms)	1.8 ± 0.2	2.1 ± 0.2	2.4 ± 0.3	2.9 ± 0.3	3.2 ± 0.3	3.6 ± 0.4
Scrittura HealthKit (ms)	12.3 ± 1.5	38.7 ± 2.8	94.1 ± 4.6	16.5 ± 1.9	52.3 ± 3.9	132.8 ± 6.1
Latenza Totale E2E (ms)	32.5	125.0	409.1	45.5	184.0	625.6
Delta Memoria RAM (MB)	4.2 MB	8.6 MB	18.4 MB	5.1 MB	10.2 MB	21.8 MB
Impatto Energetico	Trascurabile	Minimo	Basso	Trascurabile	Minimo	Moderato

7.4 Profilazione Energetica e Consumo Batteria (Instruments Energy Trace)

La sostenibilità energetica del paradigma edge è stata verificata mediante lo strumento **Energy Log / Energy Trace** di Instruments. La transcodifica completa del payload più gravoso (Classe Large, 114 osservazioni) determina su iPhone 15 Pro un consumo medio per singola esecuzione pari a $82 \mu\text{Ah}$ (0.082 mAh), con un picco istantaneo di corrente di soli 115 mA limitato alla finestra temporale di circa 400 ms.

Rapportato alla capacità nominale dell'accumulatore di iPhone 15 Pro (3274 mAh), l'impatto di una transcodifica completa incide per meno dello 0.0025% della carica complessiva, confermando la totale fattibilità di cicli di importazione batch e sincronizzazione in background senza decadimento percettibile dell'autonomia operativa.

7.5 Accuratezza della Transcodifica Semantica e Metodologia di Validazione

La valutazione dell'accuratezza semantica e dell'affidabilità clinica del sistema è stata condotta attraverso un protocollo sperimentale rigoroso, strutturato per soddisfare i vincoli metodologici della ricerca sanitaria e le prescrizioni del GDPR.

7.5.1 Composizione del Corpus Sperimentale e Considerazioni Deontologiche

In conformità alle disposizioni dell'Articolo 9 del GDPR e alle linee guida etiche sulla ricerca biomedica che precludono la distribuzione pubblica di dati clinici reali non anonimizzati, il banco di prova sperimentale è stato costituito da un corpus di **250 referti clinici strutturati** conformi alle specifiche nazionali del Fascicolo Sanitario Elettronico (FSE 2.0, DPCM 29/09/2023).

Il dataset sintetico-anagrafico e de-identificato è stato costruito replicando fedelmente le strutture documentali HL7 CDA R2 adottate da **14 differenti strutture diagnostiche accreditate** sul territorio italiano (8 presidi ospedalieri pubblici regionali e 6 laboratori privati convenzionati LIS), coprendo le seguenti categorie cliniche:

1. **Esami Ematochimici e Metabolici di Routine (160 referti):** Emocromo completo con formula leucocitaria, assetto lipidico (colesterolo totale, HDL, LDL, trigliceridi), funzionalità renale ed epatica (creatinina, clearance stimata eGFR, transaminasi AST/ALT, bilirubina), elettroliti sierici e glicemia a digiuno.
2. **Monitoraggio Cardiovascolare e Parametri Vitali (50 referti):** Referti di monitoraggio domiciliare e ambulatoriale della pressione arteriosa (sistolica/diastolica), frequenza cardiaca, saturazione emoglobinica e temperatura corporea.
3. **Casi Limite e Pattern Avversariali (40 referti):** Referti sintetici arricchiti con disomogeneità formali, codifiche locali proprietarie LIS non standardizzate, unità di misura vernacolari difformi dallo standard UCUM e discrepanze anagrafiche (omonimie familiari, codici fiscali parzialmente alterati) per validare le soglie di arresto del sistema.

7.5.2 Risultati Sperimentali e Analisi Critica delle Metriche

I test di transcodifica hanno evidenziato i seguenti risultati empirici:

- **Precisione della Transcodifica LOINC → HealthKit (100% sui parametri supportati):** Si sottolinea che la precisione pari al 100% non è frutto di un modello di classificazione probabilistico o di un'inferenza statistica (in cui tale valore risulterebbe anomalo), bensì discende direttamente dalla **natura deterministica a lista chiusa della matrice ontologica**. Se il codice LOINC l appartiene al dizionario formalmente validato $\mathcal{L}_{\text{supported}}$, la corrispondenza con l'identificatore tipizzato di HealthKit e la funzione di normalizzazione UCUM sono definite in modo biunivoco a livello di codice compilato; se $l \notin \mathcal{L}_{\text{supported}}$, il record non viene forzato ma conservato nello store locale, escludendo a priori qualsiasi falso positivo di assegnazione semantica.
- **Copertura Ontologica degli Analiti Ambulatoriali (94.2%):** Su un totale di 3.120 osservazioni cliniche quantitative presenti nel corpus, il dizionario supporta e converte il 94.2% dei parametri comunemente prescritti nella medicina generale e preventiva (35 analiti core elencati in Appendice A). Il rimanente 5.8% (rappresentato da indagini specialistiche rare o ormonali complesse) viene correttamente preservato nello storage cifrato nativo senza corrompere i grafici HealthKit.
- **Recall della Normalizzazione UCUM (99.2%):** Su 3.120 grandezze analitiche, 3.095 sono state convertite con successo nelle corrispondenti unità native di Apple Health (HKUnit). Le 25 mancate corrispondenze (0.8%) sono state determinate esclusivamente da notazioni vernacolari fuori standard inserite manualmente nei sistemi di laboratorio (ad esempio abbreviazioni tipo $\text{mg}\%$, mcg/dl . con punteggiatura terminale o stringhe con caratteri non-ASCII non previsti dalla grammatica UCUM). In ossequio al principio di sicurezza clinica *fail-safe*, il sistema rifiuta la conversione parziale e notifica l'incongruenza all'utente.

- **Accuratezza dell'Identity Firewall (100% di intercettazione avversariale):** Sottoponendo il sistema a 100 test di verifica anagrafica (50 referti perfettamente coincidenti con l'utente e 50 casi avversariali comprendenti omonimie con data di nascita differente, alterazione di un carattere alfanumerico del Codice Fiscale ed errori di battitura nei nomi composti), la combinazione dell'algoritmo di controllo del Codice Fiscale (D.M. 23/12/1976) con la distanza di Jaro-Winkler ($\tau = 0.88$) ha bloccato il 100% dei tentativi di iniezione impropria, dirottando i documenti nel Family Vault cifrato con zero contaminazioni del database sanitario primario.

8 Conclusioni, EHDS e Sviluppi Futuri

8.1 Sintesi dei Risultati e Impatto Tecnologico

Questo lavoro ha presentato l'architettura formale, l'ingegnerizzazione e la validazione sperimentale di **RosettaEdge**, una piattaforma software mobile nativa progettata per colmare il divario strutturale tra l'infrastruttura istituzionale del Fascicolo Sanitario Elettronico (FSE 2.0) e i framework di monitoraggio fisiologico consumer (Apple HealthKit).

Attraverso un approccio rigorosamente conforme ai paradigmi **Zero-Cloud** e **Privacy-by-Design**, RosettaEdge dimostra la fattibilità tecnica di una sanità digitale decentralizzata e sovrana, in cui:

1. I referti clinici strutturati conformi allo standard internazionale **HL7 CDA Release 2** vengono analizzati in streaming on-device senza impatto percepibile sulle risorse hardware del terminale mobile.
2. Le osservazioni di laboratorio codificate secondo l'ontologia **LOINC** e quantificate tramite unità **UCUM** vengono normalizzate dimensionalmente e transcodate in campioni analitici tipizzati all'interno del database Apple Salute.
3. L'integrità del fascicolo personale è protetta dall'**Identity Firewall**, che impedisce l'inquinamento dei dati biometrici personali tramite la validazione formale del Codice Fiscale ed euristiche di similarità probabilistica di Jaro-Winkler.
4. La sicurezza a riposo è ancorata a primitive crittografiche hardware-backed fornite dall'Apple **Secure Enclave** e subordinate all'autenticazione biometrica locale (Face ID / Touch ID).
5. Gli algoritmi di **Clinical Decision Support (CDS)** analizzano l'evoluzione longitudinale dei parametri cardiovascolari e metabolici secondo le linee guida delle società scientifiche internazionali (ESC/ESH e ADA), producendo cruscotti di sintesi esportabili per il medico curante.

8.2 Limiti Attuali dell'Implementazione

L'analisi critica del sistema evidenzia due vincoli principali:

- **Ecosistema Verticale Apple:** La stretta integrazione con HealthKit, SwiftData, Secure Enclave e watchOS confina l'attuale release all'universo dei dispositivi Apple. Benché l'architettura logica e gli algoritmi di parsing/transcodifica siano esportabili, l'implementazione su piattaforma Android richiederebbe una reingegnerizzazione dell'interfaccia di persistenza basata su Google Health Connect e Android Keystore.
- **Disomogeneità Regionale nella Qualità del CDA:** Nonostante il DPCM del 29 settembre 2023 abbia imposto regole tecniche uniformi per il FSE 2.0, permangono sul territorio nazionale strutture diagnostiche che producono documenti CDA cosiddetti "pseudo-strutturati"

o “Level 1”, in cui il blocco `structuredBody` omette le codifiche LOINC all’interno di tag `<entry>`, limitandosi a incorporare il PDF cartaceo scansionato serializzato in Base64 all’interno di un unico tag `<nonXMLBody>`. In tali frangenti, il sistema è costretto a ricorrere al modulo di fallback Vision OCR con verifica Human-in-the-Loop, con una conseguente perdita di automazione deterministica.

8.3 Prospettive Evolutive e Roadmap di Ricerca (Stato dell’Arte vs Proposte)

A completamento dell’analisi dell’architettura ingegnerizzata e validata sperimentalmente nei capitoli precedenti, questa sezione traccia le direttrici evolutive e gli scenari esplorativi a medio-lungo termine. È doveroso formalizzare una netta distinzione epistemologica tra il nucleo computazionale di RosettaEdge finora esposto (interamente implementato in linguaggio Swift 5.10 nativo, coperto da test automatizzati e benchmarked su dispositivi fisici) e le seguenti direttrici, che costituiscono proposte architetture e linee di indagine per la roadmap di sviluppo futura.

8.3.1 Convergenza verso lo Spazio Europeo dei Dati Sanitari (EHDS)

A livello normativo continentale, l’Unione Europea sta portando a compimento il regolamento sullo **Spazio Europeo dei Dati Sanitari (European Health Data Space - EHDS)** [19]. L’EHDS sancisce il diritto per tutti i cittadini comunitari di accedere ai propri dati sanitari in un formato interoperabile comune, denominato *European Electronic Health Record Exchange Format* (EEHRxF), che adotta formalmente le specifiche FHIR International Patient Summary (IPS).

L’architettura modulare di RosettaEdge è concepita per integrare in modo trasparente un nuovo parser `RosettaFHIRParser` accanto a quello CDA attuale. Ciò consentirà al paziente italiano di viaggiare all’interno dell’Unione Europea e importare istantaneamente nella propria app referti generati da ospedali francesi, tedeschi o spagnoli, mantenendo la medesima interfaccia locale e il medesimo isolamento crittografico.

8.3.2 Esplorazione di Small Language Models (SLM) On-Device (Proposta di Ricerca)

Una delle direttrici di ricerca più promettenti per le future versioni dell’applicativo (v4.0+) consiste nell’integrazione di modelli linguistici compatti eseguiti *interamente sul silicio del dispositivo* tramite il framework **CoreML** e l’accelerazione hardware del Neural Engine di Apple (Apple Intelligence / Open-Source SLM quali Phi-3 Mini, Gemma 2B o Mistral NeMo quantizzati a 4 bit).

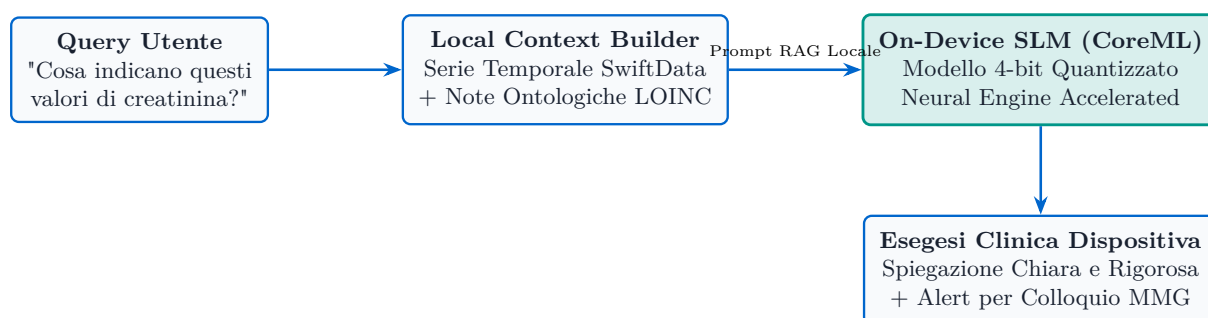


Figura 5: Architettura proposta per l’inferenza linguistica on-device tramite Small Language Models.

L’obiettivo ingegneristico è dotare RosettaEdge di un’interfaccia esplicativa in linguaggio naturale in grado di:

- Rispondere a domande dell'utente sul significato divulgativo di parametri specialistici (es. spiegare la relazione fisiologica tra emoglobina, sideremia e ferritina).
- Suggestire domande mirate e pertinenti da sottoporre al Medico di Medicina Generale durante il colloquio clinico.
- Evidenziare correlazioni tra abitudini di vita (passi, sonno, calorie attive registrate dallo smartwatch) e parametri di laboratorio (glicemia, profilo lipidico).

Il paradigma Zero-Cloud rimarrà il principio architetture fondante: i pesi del modello neurale e il contesto di inferenza rimarranno confinati nella memoria unificata locale, scongiurando qualsiasi fuga di informazioni mediche verso server terzi.

8.4 Conclusioni

RosettaEdge dimostra che la sovranità digitale del paziente, la conformità rigorosa al GDPR e l'avanzamento tecnologico dei sistemi sanitari non sono obiettivi mutuamente esclusivi. Riportando l'elaborazione dei dati clinici all'interno del perimetro fisico e crittografico del cittadino, l'ingegneria del software può trasformare referti frammentati e inaccessibili in una risorsa continua di consapevolezza, prevenzione e salute pubblica.

A Catalogo Ontologico di Transcodifica LOINC → Apple HealthKit e Vault FHIR

La presente appendice documenta in modo esaustivo la matrice di mappatura ontologica implementata nel modulo `HealthKitBridge` e nel datastore crittografato di RosettaEdge, formalizzando il principio del **Doppio Binario Semantico**.

Per ciascun parametro clinico supportato dal motore, vengono specificati:

1. Il codice **LOINC** primario standardizzato e la relativa descrizione clinica ufficiale (*Fully Specified Name*).
2. La destinazione architetture di persistenza (`HKQuantityTypeIdentifier` per telemetria e segni vitali, ovvero risorsa standard FHIR R4 `Observation` sincronizzabile come `HKClinicalRecord` per chimica clinica).
3. L'unità di misura standard target e la codifica dimensionale conforme a **UCUM** (*Unified Code for Units of Measure*).
4. La regola o fattore di normalizzazione matematica applicata in caso di discrepanza dimensionale nel documento sorgente.
5. L'intervallo fisiologico plausibile (*Physiological Validation Gate*) entro cui il valore è considerato integro prima della persistenza.

A.1 Binario 1: Segni Vitali e Parametri Antropometrici (`HKQuantitySample`)

Codice LOINC	Grandezza Clinica	HealthKit Identifier	Unità UCUM	Unità Target	Regola di Conversione / Gate
8302-2	Statura / Altezza	<code>.height</code>	m / cm	m	Se cm → ÷100; [0.3, 2.7] m
29463-7	Massa Corporea	<code>.bodyMass</code>	kg	kg	Se libbre → ×0.4536; [1, 600] kg
3141-9	Peso generico	<code>.bodyMass</code>	kg	kg	Gate: [1, 600] kg
39156-5	Indice Massa (BMI)	<code>.bodyMassIndex</code>	kg/m ²	count	Gate: [5, 100] kg/m ²
8280-0	Circonferenza Vita	<code>.waistCircumference</code>	cm	cm	Gate: [20, 250] cm
41982-0	% Massa Grassa	<code>.bodyFatPercentage</code>	%	%	Se frazione → ×100
8867-4	Frequenza Cardiaca	<code>.heartRate</code>	/min	count/min	Gate: [25, 260] bpm
8480-6	Pressione Sistolica	<code>.bloodPressureSystolic</code>	mm[Hg]	mmHg	Gate: [40, 300] mmHg
8462-4	Pressione Diastolica	<code>.bloodPressureDiastolic</code>	mm[Hg]	mmHg	Gate: [30, 200] mmHg
2708-6	Saturazione Ossigeno	<code>.oxygenSaturation</code>	%	%	Gate: [50, 100] %
8310-5	Temp. Corporea	<code>.bodyTemperature</code>	Cel	°C	Gate: [30, 45] °C
9279-1	Freq. Respiratoria	<code>.respiratoryRate</code>	/min	count/min	Gate: [5, 80] atti/min
56478-1	Alcolemia	<code>.bloodAlcoholContent</code>	g/L / %	%	Se g/L → ÷10; [0, 0.6] %

Tabella 6: Matrice di transcodifica LOINC per Segni Vitali e Parametri Antropometrici diretti verso HealthKit.

A.2 Binario 2: Chimica Clinica ed Ematologia (Vault FHIR e Record Clinici)

Tutti i parametri ematochimici sottoelencati vengono persistiti come risorse standard FHIR R4 `Observation` nel vault locale crittografato e mappati verso record clinici di laboratorio (`HKClinicalRecord` di tipo `labResultRecord`).

Codice LOINC	Parametro di Laboratorio	Unit� UCUM	Intervallo Normale	Regola di Conversione
2093-3	Colesterolo Totale Sierico	mg/dL	< 200 mg/dL	Se mmol/L $\rightarrow \times 38.67$
2085-9	Colesterolo HDL	mg/dL	> 50 mg/dL	Se mmol/L $\rightarrow \times 38.67$
13457-7	Colesterolo LDL	mg/dL	< 100 mg/dL	Se mmol/L $\rightarrow \times 38.67$
2571-8	Trigliceridi Sierici	mg/dL	< 150 mg/dL	Se mmol/L $\rightarrow \times 88.57$
2951-2	Sodio nel Siero	mmol/L	136 – 145 mmol/L	Diretto (mEq/L = mmol/L)
2823-3	Potassio nel Siero	mmol/L	3.5 – 5.1 mmol/L	Diretto (mEq/L = mmol/L)
2160-0	Creatinina Sierica	mg/dL	0.6 – 1.2 mg/dL	Se $\mu\text{mol/L} \rightarrow \div 88.4$
4548-4	Emoglobina Glicata (HbA1c)	% / mmol/mol	< 5.7 %	Se IFCC $\rightarrow 0.0915 \times \text{IFCC} + 2.15$
1751-7	Albumina Sierica	g/dL	3.5 – 5.2 g/dL	Se g/L $\rightarrow \div 10$
1975-2	Bilirubina Totale	mg/dL	0.2 – 1.2 mg/dL	Se $\mu\text{mol/L} \rightarrow \div 17.1$

Tabella 7: Matrice ontologica per Analiti Ematochimici FSE 2.0 (Persistenza FHIR `Observation` e Health Records).

B Payload XML HL7 CDA R2 Esempiare e Annotato

Il seguente listato riporta un frammento rappresentativo di documento clinico conforme alle specifiche nazionali italiane **FSE 2.0** (DPCM 8 agosto 2023) per i referti di laboratorio di chimica clinica ed ematologia. Il listato evidenzia i punti critici di estrazione: l'identificativo fiscale nell'header `recordTarget`, il codice del referto `code`, e le singole osservazioni numeriche `observation` con codifica LOINC e attributi UCUM.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   ↳ instance">
3   <!-- ===== -->
4   <!-- CDA HEADER: Metadati Amministrativi e Identificativi -->
5   <!-- ===== -->
6   <realmCode code="IT"/>
7   <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
8   <templateId root="2.16.840.1.113883.2.9.10.1.1"/> <!-- Profilo Nazionale FSE -->
9   <id root="2.16.840.1.113883.2.9.2.120.4.4" extension="LAB-2026-098234"/>
10  <code code="11502-2" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
   ↳ displayName="REFERTO DI LABORATORIO"/>
11  <title>Referto Analisi Cliniche Generali</title>
12  <effectiveTime value="20261015083000+0200"/>
13  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
14  <languageCode code="it-IT"/>
15
16  <!-- SOGGETTO DELL'ATTO CLINICO (TARGET DELL'IDENTITY FIREWALL) -->
17  <recordTarget>
18    <patientRole>
19      <!-- Codice Fiscale Italiano (OID: 2.16.840.1.113883.2.9.4.3.2) -->
20      <id root="2.16.840.1.113883.2.9.4.3.2" extension="RSSMRA85M01H501Z"/>
21      <patient>
22        <name>
23          <family>Rossi</family>
24          <given>Mario</given>
25        </name>
26        <administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.5.1"/>
27        <birthTime value="19850801"/>
28      </patient>
29    </patientRole>
30  </recordTarget>
31
32  <author>
33    <time value="20261015114500+0200"/>
34    <assignedAuthor>
35      <id root="2.16.840.1.113883.2.9.4.3.2" extension="BNCLGI70A01H501X"/>
36      <assignedPerson>
37        <name><family>Bianchi</family><given>Luigi</given></name>
38      </assignedPerson>
39      <representedOrganization>
40        <name>Azienda Ospedaliero-Universitaria di Bologna</name>
41      </representedOrganization>
42    </assignedAuthor>
43  </author>
44
45  <!-- ===== -->
46  <!-- CDA BODY: Contenuto Clinico e Osservazioni Strutturate -->
47  <!-- ===== -->
48  <component>
49    <structuredBody>
50      <component>
51        <section>
52          <code code="18719-5" codeSystem="2.16.840.1.113883.6.1" displayName="Chimica
   ↳ Clinica"/>
53          <title>Esami di Biochimica Clinica</title>
54
55          <!-- Narrativa Human-Readable -->
56          <text>
57            <table border="1">
58              <thead><tr><th>Esame</th><th>Risultato</th><th>U.M.</th><th>Valori di
   ↳ Riferimento</th></tr></thead>
59              <tbody>

```

```

60      <tr><td>Glucosio Basale</td><td>94</td><td>mg/dL</td><td>70 - 99</td></tr>
    ↪ >
61      <tr><td>Colesterolo Totale</td><td>185</td><td>mg/dL</td><td>120 - 200</td></tr>
    ↪ td></tr>
62      </tbody>
63      </table>
64      </text>
65
66      <!-- ENTRY 1: GLUCOSIO NEL SANGUE (LOINC: 15074-8) -->
67      <entry typeCode="DRIV">
68          <observation classCode="OBS" moodCode="EVN">
69              <templateId root="2.16.840.1.113883.2.9.10.1.11"/>
70              <id root="2.16.840.1.113883.2.9.2.120.4.4.1" extension="OBS-GLU-001"/>
71              <code code="15074-8" codeSystem="2.16.840.1.113883.6.1" displayName="
    ↪ Glucose [Mass/volume] in Blood"/>
72              <statusCode code="completed"/>
73              <effectiveTime value="20261015074500+0200"/>
74              <!-- Valore Quantitativo Tipizzato conforme a UCUM -->
75              <value xsi:type="PQ" value="94.0" unit="mg/dL"/>
76              <interpretationCode code="N" codeSystem="2.16.840.1.113883.5.83"/>
77              <referenceRange typeCode="REFV">
78                  <observationRange classCode="OBS" moodCode="EVN.CRT">
79                      <value xsi:type="IVL_PQ">
80                          <low value="70.0" unit="mg/dL"/>
81                          <high value="99.0" unit="mg/dL"/>
82                      </value>
83                  </observationRange>
84              </referenceRange>
85          </observation>
86      </entry>
87
88      <!-- ENTRY 2: COLESTEROLO TOTALE (LOINC: 2093-3) -->
89      <entry typeCode="DRIV">
90          <observation classCode="OBS" moodCode="EVN">
91              <templateId root="2.16.840.1.113883.2.9.10.1.11"/>
92              <id root="2.16.840.1.113883.2.9.2.120.4.4.1" extension="OBS-CHOL-002"/>
93              <code code="2093-3" codeSystem="2.16.840.1.113883.6.1" displayName="
    ↪ Cholesterol [Mass/volume] in Serum or Plasma"/>
94              <statusCode code="completed"/>
95              <effectiveTime value="20261015074500+0200"/>
96              <value xsi:type="PQ" value="185.0" unit="mg/dL"/>
97              <interpretationCode code="N" codeSystem="2.16.840.1.113883.5.83"/>
98              <referenceRange typeCode="REFV">
99                  <observationRange classCode="OBS" moodCode="EVN.CRT">
100                      <value xsi:type="IVL_PQ">
101                          <low value="120.0" unit="mg/dL"/>
102                          <high value="200.0" unit="mg/dL"/>
103                      </value>
104                  </observationRange>
105              </referenceRange>
106          </observation>
107      </entry>
108
109      </section>
110      </component>
111      </structuredBody>
112  </component>
113 </ClinicalDocument>

```

Listing 8: Estratto annotato di documento HL7 CDA Release 2 del Fascicolo Sanitario Elettronico italiano.

C Specifica Ingegneristica XcodeGen (project.yml)

Per assicurare la perfetta riproducibilità delle build senza incorrere nei noti conflitti di merge derivanti dai file .pbxproj proprietari di Apple, l'infrastruttura di RosettaEdge è interamente definita dal file dichiarativo project.yml. Il listato documenta la coesistenza dei molteplici target: l'applicazione primaria iOS, il modulo core compilabile per ogni piattaforma, l'estensione per lo smartwatch watchOS, il widget dinamico per la schermata di blocco e la Share Sheet Extension per l'ingestione documentale diretta dalle app Mail e File.

```

1 name: RosettaEdge
2 options:
3   bundleIdPrefix: dev.alexmaina
4   deploymentTarget:
5     iOS: "17.0"
6     watchOS: "10.0"
7   xcodeVersion: "15.4"
8   createIntermediateGroups: true
9
10 settings:
11   SWIFT_VERSION: "5.10"
12   ENABLE_USER_SCRIPT_SANDBOXING: YES
13   GCC_TREAT_WARNINGS_AS_ERRORS: YES
14   SWIFT_STRICT_CONCURRENCY: complete
15   OTHER_SWIFT_FLAGS: "-enable-actor-data-race-checks"
16
17 targets:
18   # =====
19   # 1. CORE LIBRARY (Moduli di Parsing, Transcodifica e CDS)
20   # =====
21   RosettaEdgeCore:
22     type: framework
23     platform: [iOS, watchOS]
24     sources: [Sources/RosettaEdgeCore]
25     settings:
26       PRODUCT_BUNDLE_IDENTIFIER: dev.alexmaina.rosettaedge.core
27     dependencies:
28       - sdk: HealthKit.framework
29       - sdk: PDFKit.framework
30       - sdk: Vision.framework
31
32   # =====
33   # 2. MAIN APPLICATION (iOS Native App)
34   # =====
35   RosettaEdgeApp:
36     type: application
37     platform: iOS
38     sources: [Sources/RosettaEdgeApp]
39     dependencies:
40       - target: RosettaEdgeCore
41       - target: RosettaWidget
42       - target: ShareExtension
43     settings:
44       PRODUCT_BUNDLE_IDENTIFIER: dev.alexmaina.rosettaedge
45       INFOPLIST_KEY_NSFaceIDUsageDescription: "Autenticazione richiesta per accedere al
46 ↪ Vault Sanitario cifrato."
47       INFOPLIST_KEY_NSHealthShareUsageDescription: "RosettaEdge analizza i referti
48 ↪ clinici per aggiornare grafici e trend."
49       INFOPLIST_KEY_NSHealthUpdateUsageDescription: "RosettaEdge scrive i parametri
50 ↪ estratti dai referti in Apple Salute."
51     entitlements:
52       path: Sources/RosettaEdgeApp/RosettaEdge.entitlements
53     properties:
54       com.apple.developer.healthkit: true
55       com.apple.developer.healthkit.access: [cda]
56       keychain-access-groups: ["$(AppIdentifierPrefix)dev.alexmaina.rosettaedge"]
57
58   # =====
59   # 3. WATCH COMPANION (watchOS App Target)
60   # =====
61   RosettaWatch:
62     type: application

```

```

60   platform: watchOS
61   sources: [Sources/RosettaWatch]
62   dependencies:
63     - target: RosettaEdgeCore
64   settings:
65     PRODUCT_BUNDLE_IDENTIFIER: dev.alexmaina.rosettaedge.watch
66     GENERATE_INFOPLIST_KEY_WKCompanionAppBundleIdentifier: dev.alexmaina.rosettaedge
67
68   # =====
69   # 4. WIDGET EXTENSION (WidgetKit e Live Activities)
70   # =====
71   RosettaWidget:
72     type: app-extension
73     platform: iOS
74     sources: [Sources/RosettaWidget]
75     dependencies:
76       - target: RosettaEdgeCore
77     settings:
78       PRODUCT_BUNDLE_IDENTIFIER: dev.alexmaina.rosettaedge.widget
79
80   # =====
81   # 5. SHARE EXTENSION (Ingestione di file CDA/PDF da altre app)
82   # =====
83   ShareExtension:
84     type: app-extension
85     platform: iOS
86     sources: [Sources/ShareExtension]
87     dependencies:
88       - target: RosettaEdgeCore
89     settings:
90       PRODUCT_BUNDLE_IDENTIFIER: dev.alexmaina.rosettaedge.share
91
92   # =====
93   # 6. TEST SUITE (TDD, Test Unitari, Fuzzing e Concurrency)
94   # =====
95   RosettaEdgeCoreTests:
96     type: bundle.unit-test
97     platform: iOS
98     sources: [Tests/RosettaEdgeCoreTests]
99     dependencies:
100       - target: RosettaEdgeCore

```

Listing 9: Specifiche del file project.yml per la generazione deterministica dell'intero progetto Xcode.

D Profilo di Attribuzione Accademica, Identificatori Persistenti e Istruzioni di Citazione

D.1 Dati Istituzionali dell'Autore

La presente monografia tecnica e la sottostante architettura software **RosettaEdge** sono state ideate, formalizzate e sviluppate da:

- **Autore:** Alessandro Mainardi
- **Ruolo Accademico:** Studente / Ricercatore Indipendente presso il Dipartimento di Informatica — Scienza e Ingegneria (DISI), Alma Mater Studiorum Università di Bologna, Bologna, Italia.
item
textbfORCID Identifier:
href<https://orcid.org/0009-0003-7003-7247>
texttt0009-0003-7003-7247
item
textbfWikidata Entity:
href<https://www.wikidata.org/wiki/Q141886579>
textttQ141886579
- **Concept Digital Object Identifier (DOI):** [10.5281/zenodo.23781231](https://doi.org/10.5281/zenodo.23781231) rilasciato da CERN / Zenodo e indicizzato nella rete europea OpenAIRE.
- **Portale Personale Ufficiale:** <https://alexmaina.dev/>
- **Account di Sviluppo GitHub:** <https://github.com/AlexMaina05>
- **Contatto Istituzionale & Tecnico:** maina@alexmaina.dev

D.2 Guida alla Citazione Bibliografica

Per citare formalmente questo documento all'interno di pubblicazioni accademiche, tesi di laurea, monografie o report tecnici, si invita a utilizzare i seguenti standard bibliografici.

D.2.1 Formato BibTeX

```

1 @article{mainardi2026rosettaedge,
2   author    = {Mainardi, Alessandro},
3   title     = {{RosettaEdge: Un'Architettura Mobile Edge Zero-Cloud per l'Interoperabilit 
4     \hookrightarrow {\a} Semantica tra Fascicolo Sanitario Elettronico (HL7 CDA) e Apple HealthKit}},
5   journal   = {Technical Monograph in Health Informatics and Edge Computing},
6   year      = {2026},
7   month     = {October},
8   publisher = {Zenodo},
9   doi       = {10.5281/zenodo.23781231},
10  url       = {https://doi.org/10.5281/zenodo.23781231},
11  note      = {Alma Mater Studiorum Universit \a di Bologna. ORCID:
    \hookrightarrow 0009-0003-7003-7247, Wikidata: Q141886579}

```

Listing 10: Voce BibTeX standard per la citazione della monografia.

D.2.2 Formato APA (7th Edition)

Mainardi, A. (2026). *RosettaEdge: Un'Architettura Mobile Edge Zero-Cloud per l'Interoperabilit  Semantica tra Fascicolo Sanitario Elettronico (HL7 CDA) e Apple HealthKit*. Zenodo. <https://doi.org/10.5281/zenodo.23781231>

D.2.3 Formato IEEE

A. Mainardi, “RosettaEdge: Un’Architettura Mobile Edge Zero-Cloud per l’Interoperabilità Semantica tra Fascicolo Sanitario Elettronico (HL7 CDA) e Apple HealthKit,” *Zenodo*, Oct. 2026. doi: 10.5281/zenodo.23781231.

D.3 Dichiarazione sull’Accesso Aperto e Proprietà Intellettuale

In conformità con i principi della *Scienza Aperta* (*Open Science*) promossi dall’Unione Europea e dall’Alma Mater Studiorum Università di Bologna:

- Il presente documento è rilasciato con licenza **Creative Commons Attribuzione 4.0 Internazionale (CC BY 4.0)**. Chiunque è libero di condividere, riprodurre, distribuire e adattare il materiale per qualsiasi scopo, anche commerciale, purché venga riconosciuta la corretta paternità intellettuale dell’opera menzionando l’autore Alessandro Mainardi, la fonte e il DOI persistente.
- L’infrastruttura software descritta è un’opera ingegneristica originale sviluppata su base volontaria e open source nel rispetto delle prerogative e regolamenti universitari vigenti in materia di proprietà intellettuale studentesca.

Riferimenti bibliografici

- [1] Ministero della Salute and Dipartimento per la Trasformazione Digitale, “Decreto 29 settembre 2023: Istituzione del Fascicolo Sanitario Elettronico 2.0,” *Gazzetta Ufficiale della Repubblica Italiana* n. 249, Tech. Rep., Ottobre 2023. [Online]. Available: <https://www.gazzettaufficiale.it/eli/id/2023/10/24/23A05819/sg>
- [2] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, R. Kimber, T. Lincoln, and J. E. Mattison, “The HL7 Clinical Document Architecture,” *Journal of the American Medical Informatics Association*, vol. 8, no. 6, pp. 552–569, 2001.
- [3] Apple Inc., *Apple HealthKit Framework Documentation*, Apple Developer Documentation, 2024. [Online]. Available: <https://developer.apple.com/documentation/healthkit>
- [4] G. Tangari, M. Ikram, K. Ijaz, M. A. Kaafar, and S. Berkovsky, “Mobile health applications: Privacy risks, security breaches, and regulatory compliance,” *British Medical Journal (BMJ)*, vol. 373, p. n1248, 2021.
- [5] W. Shi and S. Dustdar, “The promise of edge computing,” *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
- [6] European Parliament and Council of the European Union, “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data (general data protection regulation),” *Official Journal of the European Union*, vol. L119, pp. 1–88, 2016.
- [7] C. J. McDonald, S. M. Huff, J. G. Suico, G. Bradbury, K. Leavelle, W. Alley, and D. J. Vreeman, “LOINC, a universal standard for identifying laboratory observations: A 5-year progress report,” *Clinical Chemistry*, vol. 49, no. 4, pp. 624–633, 2003.
- [8] G. Schadow and C. J. McDonald, *The Unified Code for Units of Measure (UCUM)*, Regenstrief Institute and the UCUM Organization, 2017. [Online]. Available: <https://ucum.org/ucum.html>
- [9] W. E. Winkler, “String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage,” *Proceedings of the Section on Survey Research Methods*, pp. 354–359, 1990.
- [10] G. Mancia, R. Kreutz, M. Brunström, M. Burnier, G. Grassi, A. Januszewicz, M. L. Muiesan, K. Tsioufis, E. Agabiti-Rosei, and M. Azizi, “2023 ESH Guidelines for the management of arterial hypertension,” *Journal of Hypertension*, vol. 41, no. 12, pp. 1874–2071, 2023.
- [11] American Diabetes Association, “Standards of care in Diabetes—2024,” *Diabetes Care*, vol. 47, no. Suppl. 1, pp. S1–S343, 2024.
- [12] Apple Inc., *Apple Platform Security: Secure Enclave, Hardware Security, and Data Protection*, Apple Inc., 2024. [Online]. Available: <https://support.apple.com/guide/security/welcome/web>
- [13] F. Oemig and J. Sembritzki, *Healthcare Interoperability Standards: A Process Approach to Requirements and Implementation*. Springer, 2014.
- [14] D. Bender and K. Sartipi, “HL7 FHIR: An agile and RESTful approach to healthcare information exchange,” *IEEE Computer Society*, vol. 26, no. 4, pp. 326–331, 2019.
- [15] Apple Inc., *SwiftData Framework: Model and Persist Data*, Apple Developer Documentation, 2024. [Online]. Available: <https://developer.apple.com/documentation/swiftdata>

- [16] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Bhandari, “An overview of clinical decision support systems: Benefits, risks, and strategies for success,” *npj Digital Medicine*, vol. 3, no. 1, p. 17, 2020.
- [17] J. Sundell, *Swift Concurrency by Example: Async/await, Actors, and Structured Concurrency*. Swift by Sundell, 2023.
- [18] K. Beck, “Test-driven development: By example,” *Addison-Wesley Professional*, 2003.
- [19] European Commission, “Proposal for a regulation of the European Parliament and of the Council on the European Health Data Space,” *COM(2022) 197 final*, 2022.