



Consiglio Nazionale delle Ricerche  
Istituto di Calcolo e Reti ad Alte Prestazioni

## Analisi e valutazione di strumenti utili alla progettazione di una piattaforma IoT abilitante in ambito Wellness

*Iolanda Apa, Agostino Forestiero*

**RT- ICAR-CS-23-07**

**Settembre 2023**



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)  
– Sede di Cosenza, Via P. Bucci 8-9C, 87036 Rende, Italy, URL: [www.icar.cnr.it](http://www.icar.cnr.it)  
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: [www.icar.cnr.it](http://www.icar.cnr.it)  
– Sezione di Palermo, Via Ugo La Malfa, 153, 90146 Palermo, URL: [www.icar.cnr.it](http://www.icar.cnr.it)

# SOMMARIO

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Piattaforme</b>	<b>4</b>
2.1	WSO2	4
2.2	Fiware	6
2.3	IBM Watson	7
2.4	Amazon AWS IoT	8
2.5	Microsoft Azure	8
2.6	I dispositivi indossabili	9
2.7	Stato dell'arte	10
<b>3</b>	<b>Health/Fitness kit Connector</b>	<b>13</b>
3.1	Google Fit	13
3.2	The Fitness Store	14
3.3	The sensor framework	14
3.4	Permissions and user controls	14
3.5	Google Fit APIs	14
3.6	Samsung Digital Health Platform (SDH)	14
3.7	Apple HealthKit	15
3.8	Open-mhealth	16
3.8.1	Standardizzare i dati	16
3.8.2	Memorizzazione dei dati	16
3.8.3	Integrazione di flussi	16
3.9	Community Health toolkit	17
3.10	Monitoraggio con nodo-edge	17
<b>4</b>	<b>Node-RED: a flow-based development tool</b>	<b>20</b>
4.1	I flussi	21
4.2	Node-RED e l'Internet of Things	25
<b>5</b>	<b>Conclusioni</b>	<b>26</b>
	<b>Bibliografia</b>	<b>27</b>

# 1 INTRODUZIONE

Lo scopo del presente documento è quello di riportare i risultati del processo di analisi e di valutazione degli strumenti disponibili per la progettazione e definizione di una piattaforma IoT abilitante, effettuato nelle attività del progetto K4H - Kit for Health, POR CALABRIA FESR-FSE 2014-2020, ASSE I – PROMOZIONE DELLA RICERCA E DELL'INNOVAZIONE, Obiettivo specifico 1.2 "Rafforzamento del sistema innovativo regionale e nazionale", Azione 1.2.2 "Supporto alla realizzazione di progetti complessi di attività di ricerca e sviluppo su poche aree tematiche di rilievo e all'applicazione di soluzioni tecnologiche funzionali alla realizzazione delle strategie di S3".

L'obiettivo principale del progetto, infatti, è la realizzazione di una piattaforma, basata su tecnologie proprie dell'Internet of Things, per la raccolta e l'elaborazione di dati e informazioni relativi a parametri indicativi dello stato di wellness e Health di una persona. La piattaforma progettata, abiliterà un ecosistema di comunicazione tra persone, oggetti e servizi, ed inoltre, attraverso moduli componibili e configurabili, fornirà gli elementi necessari per la costruzione di soluzioni verticali e orizzontali, scalabili e flessibili basati sull'interazione e cooperazione di oggetti connessi fra loro. A tale scopo, inizialmente è stato effettuato uno studio dello stato dell'arte dei componenti strutturali e funzionali e successivamente dei dispositivi smart e delle tecnologie abilitanti propri dell'IoT.

Il progetto di ricerca mira al rilascio del prototipo di una piattaforma web-based composta da: i) un sottosistema IoT concernente prevalentemente l'acquisizione di dati da sensori e la comunicazione tra i device interconnessi; ii) un engine per l'analisi dei dati raccolti in merito ai parametri di interesse individuati (es. numero di passi effettuati, ecc.); iii) un modulo di cybersecurity per la tutela dei dati degli utenti.

Il processo di costruzione del prototipo condurrà, inoltre, all'ottenimento di una serie di soluzioni tecnologiche intermedie di raccolta, organizzazione, data analysis di dati clinici, ambientali, sociali e di life style di sportivi, amatori, cittadini. Gli outcome della piattaforma saranno d'interesse per le imprese industriali e di servizi che si rivolgeranno sempre di più a un target costituito da coorti di cittadini molto numerosi e mediamente a reddito più elevato delle generazioni successive, ma anche per i decisori pubblici per le politiche di welfare dedicato.

Il documento è strutturato come segue: la sezione 2 introdurrà le principali piattaforme incentrate sull'utilizzo del paradigma dell'Internet of Things come WSO2, FIWARE, IBM WATSON, ecc. che sono state analizzate e valutate al fine di selezionare quella più adeguata all'utilizzo nella piattaforma K4H. La sezione 3 riporta l'analisi e la valutazione dei sistemi cosiddetti CONNECTOR, ovvero sistemi sensor-based necessari per il monitoraggio degli apparati fisici che sono stati valutati ed integrati nella piattaforma oggetto del progetto. Particolare attenzione è stata data all'analisi di Node-RED, un potente strumento per la costruzione di applicazioni Internet of Things (IoT) per semplificare il "wiring together" dei blocchi di codice per l'esecuzione dei task. La sezione 4 riporta lo studio effettuato sulle piattaforme di analisi dei dati recuperati dagli apparati sensoristici. I sistemi principali analizzati e selezionati per la piattaforma prototipale sono basati sul framework HADOOP/MAP REDUCE. Infine la sezione 5 riporta il progetto dell'architettura del motore della piattaforma K4H realizzata utilizzando gli strumenti analizzati nelle parti precedenti.

L'Internet of Things (IoT) fornisce una vista virtuale, tramite Internet Protocol, a una grande varietà di oggetti della vita reale, che vanno da un'auto, a una tazza di caffè, a un edificio, agli alberi in una foresta, alle persone. La cosa interessante è che si può accedere allo stato e alla posizione di qualsiasi "cosa" a cui si è essere interessati [7].

Il software IoT è necessario non solo per disporre di enormi volumi di dati eterogenei in tempo reale, ma anche per supportare diverse applicazioni complesse. Un approccio ontologico, ne potrebbe essere un esempio: sulla base di un modello di informazione astratto, potrebbero essere usate informazioni che a loro volta incapsulano, compongono, scompongono, trasferiscono, tracciano e interagiscono nella gestione del ciclo di vita di un prodotto. Combinando ontologia e software si potrebbero creare piattaforme che forniscono una base di supporto alle informazioni sia per l'integrazione dei dati che per l'interazione intelligente [6].

I dati generati che possono essere considerati come un punto di collegamento chiave tra cose e oggetti su Internet sono veramente tanti. Una molteplicità di sensori interagisce attraverso le reti wireless, raccogliendo dati e offrendo servizi che vanno dal monitoraggio dell'inquinamento ambientale, alla gestione e al ripristino delle catastrofi, al miglioramento della qualità della vita nelle case, al funzionamento delle città intelligenti. Tuttavia, nonostante i benefici percepiti che stiamo realizzando da questi sensori, l'alba IoT presenta nuove sfide. Una delle grandi sfide del mondo IoT riguarda la progettazione dell'infrastruttura appropriata per acquisire e archiviare l'enorme quantità di dati da sensori eterogenei, in modo tale che gli utenti possano cercare, trovare e utilizzare senza problemi i loro dati. Alcuni studi hanno cercato di affrontare il problema realizzando un'architettura IoT integrata che combina le funzionalità di Service-Controlled Networking (SCN) con il cloud computing [8]. Altri studi sono stati focalizzati nell' edge computing per ridurre l'enorme quantità di dati provenienti dai dispositivi IoT verso Internet proponendo BodyEdge [9], una nuova architettura che ben si adatta alle applicazioni mediche nel contesto del settore sanitario emergente. I vantaggi della piattaforma software progettata sono stati valutati in termini di riduzione della larghezza di banda e dei tempi di elaborazione attraverso un banco di prova reale su diverse piattaforme hardware.

## 2 PIATTAFORME

Attualmente le applicazioni e le piattaforme sono strettamente collegate, impedendo la loro interazione con altre applicazioni e piattaforme, i sensori e gli attuatori comunicano solo all'interno di un sistema, alcune piattaforme non implementano alcuni servizi importanti (ad es. Scoperta), o lo fanno in modo incompatibile. Gli elementi mobili possono essere mancanti o inaccessibili. Il software per dispositivi IoT non è mai indipendente dalla piattaforma poiché le aziende creano software proprietario. Questi fatti presentano enormi difficoltà per il raggiungimento dell'interoperabilità.

### 2.1 WSO2

WSO2 è un middleware di architettura orientata ai servizi (SOA) open source. È progettato con componenti indipendenti, quindi può essere adattato per una soluzione snella e mirata per le applicazioni aziendali. L'intero stack di middleware WSO2 funziona senza soluzione di continuità su cloud privati, pubblici, gestiti da WSO2 e ibridi, nonché on-premise.

Per proteggere completamente dal blocco, tutti i prodotti WSO2 sono open source al 100% e basati su standard aperti. Inoltre, i prodotti WSO2 rilasciati con licenza Apache versione 2.0. WSO2 è aperto a chiunque sia interessato ai propri prodotti per essere coinvolto nella comunità WSO2. Gli sviluppatori possono estendere la piattaforma, personalizzare il codice e utilizzare qualsiasi modello di programmazione che preferiscono, segnalare bug o vulnerabilità della sicurezza, preparare materiali di formazione, partecipare a forum ed eventi, iscriversi a mailing list pubbliche, ecc.

I prodotti WSO2 fanno ampio uso della tecnologia Java e sono basati su WSO2 Carbon, la piattaforma middleware SOA dell'azienda. Carbon utilizza Apache Axis2 e incapsula funzionalità SOA come servizi dati, gestione dei processi aziendali, routing / trasformazione ESB, regole, sicurezza, limitazione, memorizzazione nella cache, registrazione e monitoraggio.

Non tutti i componenti vengono utilizzati come implementazioni autonome. Molti di questi vengono utilizzati per integrare le capacità o aggiungere funzionalità a un'implementazione del bus di servizio aziendale. I componenti principali che possono essere utilizzati nel middleware WSO2 sono:

- **Gestione API**
  - API Manager: piattaforma di gestione API per la creazione, la distribuzione e la gestione di API per esporre dati e funzionalità dei sistemi di back-end.
  - API Cloud: servizio di gestione API ospitato.

- **Integrazione**
  - Enterprise Service Bus: consente agli sviluppatori di connettersi e gestire sistemi e software in conformità con i principi di governance SOA.
  - Server dei servizi dati: fornisce un'interfaccia del servizio Web per gli archivi dati.
  - Message Broker: traduce, convalida e instrada i messaggi tra i sistemi.
  - Business Process Server: una console grafica per la gestione dei processi aziendali e delle attività umane.
- **Analitica**
  - Data Analytics Server: analisi in tempo reale, batch, interattive e predittive che utilizzano dati aziendali.
  - Elaboratore di eventi complessi: elaborazione e rilevamento di eventi in tempo reale. Identificare i modelli da più origini dati, analizzarne l'impatto. Utilizza WSO2 Siddhi e Apache Storm.
  - Machine Learner: analisi dei dati esplorativi utilizzando modelli per generare previsioni. Utilizza Apache Spark.
- **Gestione dell'identità e sicurezza**
  - Identity Server: collega e gestisce più identità tra applicazioni, API, dispositivi cloud, mobili e Internet of Things.
- **Servizi e app Dev**
  - Server applicazioni: consente di condividere logiche, dati e processi aziendali nell'intero ecosistema IT. Fornisce hosting di applicazioni SaaS condivise, multi-tenant e scalabili in modo elastico.
  - App Cloud: fornisce un ecosistema cloud completo che si evolve per consentire una facile ed efficiente integrazione, identità e gestione API per la tua impresa digitale.
  - Microservices Framework per Java: consente di creare microservizi in Java con distribuzione basata su container.
- **Gestione e governance**
  - App Manager: facilita il processo di creazione, distribuzione e gestione delle applicazioni.
  - Registro della governance: archiviazione, catalogazione, indicizzazione, gestione e governance dei metadati relativi alle risorse aziendali.
- **Mobile e IoT**
  - IoT Server: piattaforma Internet of Things per la gestione dei dispositivi.
  - Enterprise Mobility Manager: gestione dei dispositivi e applicazione delle politiche aziendali per i dispositivi mobili.

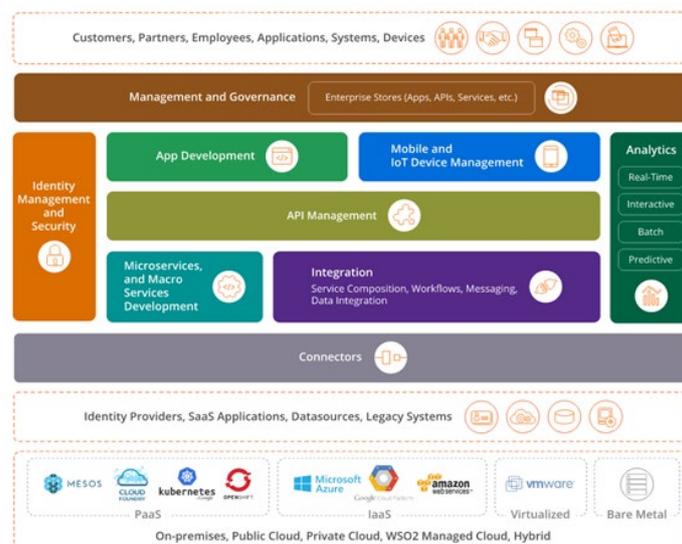


FIGURA 1 WSO2 ARCHITECTURE

## 2.2 FIWARE

FIWARE è una piattaforma middleware, guidata dall'Unione Europea nell'ambito del Programma di partenariato pubblico-privato per il futuro di Internet, per lo sviluppo e l'implementazione globale di applicazioni intelligenti per Internet futuro in molteplici settori verticali.

La piattaforma FIWARE fornisce un set piuttosto semplice ma potente di API (Application Programming Interface) che facilita lo sviluppo di Smart Applications in più settori verticali. Le specifiche di queste API sono pubbliche e esenti da royalty. Inoltre, un'implementazione di riferimento open source di ciascuno dei componenti di FIWARE è disponibile al pubblico in modo che più fornitori di FIWARE possano emergere più rapidamente sul mercato con una proposta a basso costo.

I principali risultati di FIWARE saranno un'architettura aperta e un'implementazione di riferimento di una nuova infrastruttura di servizi, basata su blocchi generici e riutilizzabili sviluppati in precedenti progetti di ricerca.

FIWARE si basa sulle seguenti basi principali:

- **Service Delivery Framework:** infrastruttura per la creazione, la pubblicazione, la gestione e il consumo di servizi FIWARE lungo tutto il loro ciclo di vita, affrontando tutti gli aspetti tecnici e commerciali
- **Cloud Hosting:** rappresenta il livello che fornisce le risorse di calcolo, archiviazione e rete, su cui vengono forniti e gestiti i servizi
- **Servizi di supporto:** rappresentano le strutture per accedere, elaborare e analizzare in modo efficace enormi flussi di dati e classificarli semanticamente in preziose conoscenze
- **Abilitazione IoT:** rappresenta un bridge attraverso il quale i servizi FIWARE si interfacciano e sfruttano l'ubiquità dei dispositivi eterogenei e caratterizzati da limitate risorse
- **Interfaccia alle reti:** interfacce aperte per reti e dispositivi, garantendo la connettività a servizi, di cui si ha bisogno, fruibili attraverso la piattaforma
- **Sicurezza:** meccanismi che garantiscono che la fornitura e l'utilizzo dei servizi siano affidabili e soddisfino i requisiti di sicurezza e privacy

I Generic Enabler (GE) di FIWARE sono raggruppati e organizzati in capitoli. Ogni capitolo fornisce una serie di GE che lavorano e comunicano insieme per fornire supporto alle seguenti aree (consultare il catalogo FIWARE<sup>1</sup>):

- Data / Context
- IoT
- Interfaccia utente avanzata
- Sicurezza
- Interfaccia con reti e dispositivi (I2ND)
- Apps
- Cloud

FIWARE fornisce inoltre Enabler specifici di dominio rivolti a fornire funzionalità e API per questi domini:

- Manufacturing
- Trasporto, logistica e prodotti alimentari
- Mobilità personale
- TV social, servizi di città mobili e videogiochi
- Città intelligenti e pubblica sicurezza
- eHealth
- Energia intelligente

---

<sup>1</sup> <http://catalogue.fiware.org/>

● Ambiente

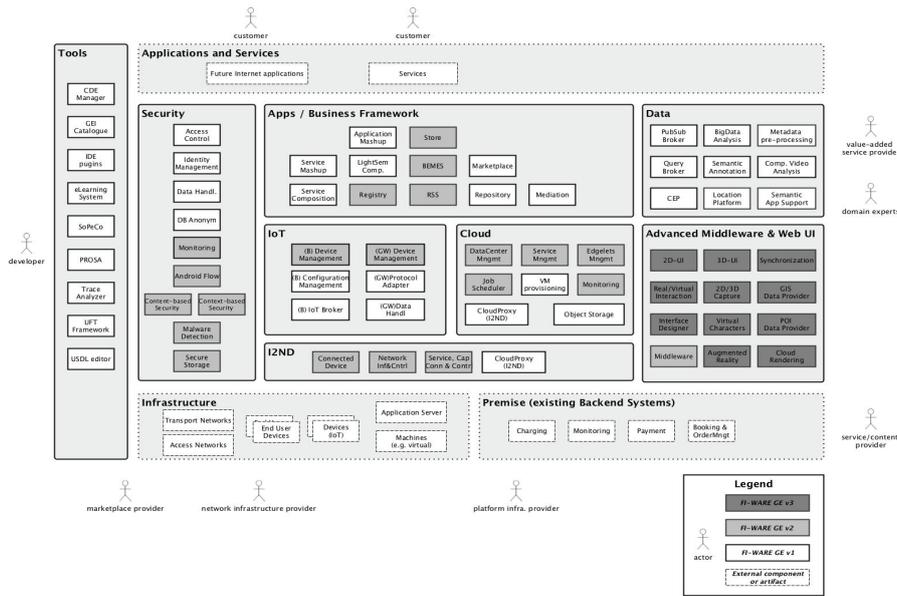


FIGURA 2 ARCHITETTURA GENERICA FIWARE

## 2.3 IBM WATSON

Internet of Things su Bluemix utilizza la piattaforma IBM Watson IoT. In generale, Bluemix funziona sia come una piattaforma cloud che servizio (PaaS) alimentata da progetti open source e sviluppata da IBM. Supporta più linguaggi di programmazione come Java, Node.js, Go, PHP, Python, Ruby Sinatra, Ruby on Rails e può essere esteso per supportare altri linguaggi come Scala utilizzando buildpack. Supporta inoltre più servizi e DevOps integrati per creare, eseguire, distribuire e gestire applicazioni sul cloud. Bluemix si basa sulla tecnologia aperta Cloud Foundry e funziona su infrastruttura SoftLayer. Esistono piani gratuiti iniziali che includono fino a 20 dispositivi, 10 associazioni di applicazioni e 100 MB di scambio di dati. L'utilizzo aggiuntivo viene fatturato a una tariffa per MB.

Oltre ai comuni servizi IoT, Bluemix fornisce estensioni per Business Rules, elaborazione Hadoop, livello di database NoSQL Cloudant e MongoDB, diversi strumenti DevOps, messaggistica, analisi GeoSpatial e accesso ai servizi Watson, in particolare per l'elaborazione del linguaggio naturale.

La connessione alla piattaforma è possibile per dispositivi e gateway. I dati sono protetti nel cloud connettendosi tramite protocollo di messaggistica MQTT o HTTP. Watson è l'hub che consente l'installazione e la gestione di dispositivi e applicazioni collegati che consentono l'accesso a dati storici e in tempo reale. Sono disponibili API di riposo e in tempo reale per facilitare le connessioni tra dispositivi e applicazioni.

## 2.4 AMAZON AWS IoT

AWS IoT è una piattaforma cloud gestita che consente ai dispositivi connessi di interagire facilmente e in modo sicuro con le applicazioni cloud e altri dispositivi. Fornisce servizi ospitati nei principali servizi cloud AWS (acronimo di Amazon Web Services) e sfrutta la maggior parte dei servizi / moduli per facilitare alcuni dei processi più comuni come

- AWS Lambda (cloud computing senza server)
- Kinesis (operazioni di streaming dei dati)
- S3 (cloud storage)
- Apprendimento automatico
- DynamoDB (database NoSQL)
- CloudWatch (monitoraggio delle applicazioni cloud)
- CloudTrail (registrazione chiamate API)
- Servizio Elasticsearch con integrazione Kibana integrata (visualizzazione dati)

Il modulo IoT AWS copre esclusivamente quegli aspetti che sono esclusivi (o fortemente vincolati) del dominio IoT (vedere la documentazione disponibile):

- Gestione dei dispositivi
- Connettori SDK del dispositivo
- Message Broker
- Entità virtuali (ombre del dispositivo)
- Motore di regole

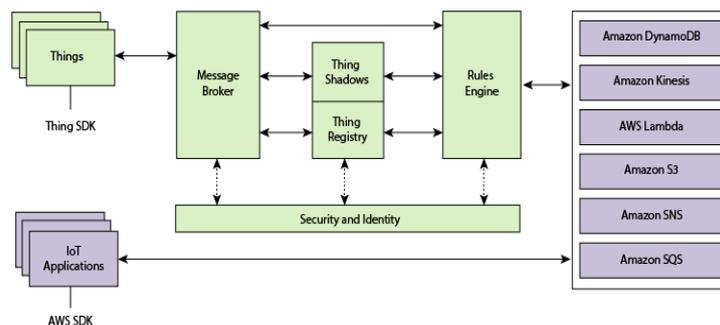


FIGURA 3 ARCHITETTURA AMAZON AWS IoT

## 2.5 MICROSOFT AZURE

L'hub IoT di Azure è un'estensione IoT integrato nel cloud di Microsoft Azure. Il suo scopo principale è consentire comunicazioni bidirezionali affidabili e sicure tra un gran numero di dispositivi IoT e un motore back-end, tipicamente ospitato su cloud. L'hub IoT di Azure offre messaggistica affidabile da dispositivo a cloud e da cloud a dispositivo, comunicazioni sicure usando credenziali di sicurezza per dispositivo e controllo degli accessi. Offre un monitoraggio completo per la connettività dei dispositivi e gli eventi di gestione dell'identità dei dispositivi e include librerie di dispositivi per le lingue e le piattaforme più popolari. Fornisce inoltre un gateway IoT SDK per l'elaborazione dello sviluppo e la logica dell'applicazione al limite.

La piattaforma IoT di Microsoft Azure è composta da servizi di piattaforma di base e componenti a livello di applicazione per facilitare le esigenze di elaborazione in tre aree principali di una tipica soluzione IoT. Ciò comprende

- Connettività del dispositivo
- Elaborazione, analisi e gestione dei dati
- Presentazione e connettività aziendale.

I dispositivi possono essere collegati direttamente o indirettamente tramite un gateway ed entrambi possono implementare edge intelligence con diversi livelli di capacità di elaborazione. Un gateway cloud fornisce endpoint per la connettività dei dispositivi e facilita la comunicazione bidirezionale con il sistema di backend. Il back-end comprende più componenti per fornire la registrazione e il rilevamento dei dispositivi, la raccolta, la trasformazione e l'analisi dei dati, nonché la logica e le visualizzazioni aziendali. Il livello di integrazione e presentazione aziendale è responsabile dell'integrazione dell'ambiente IoT nei processi aziendali di un'azienda.

Per quanto riguarda la connettività, l'hub di Microsoft Azure supporta diverse opzioni di connettività al fine di integrare le risorse IoT, che possono essere connesse direttamente o indirettamente tramite i cosiddetti gateway di campo. Il principale punto di integrazione verso i dispositivi fornisce è l'hub IoT di Azure che offre supporto per tre protocolli:

- AMQP
- MQTT
- Protocolli HTTP 1.1 su TLS

L'SDK del dispositivo Azure IoT può essere usato per semplificare lo sviluppo di client IoT che possono connettersi all'hub IoT di Azure tramite le opzioni sopra. I dispositivi più vincolati richiedono un'implementazione del gateway di campo per la traduzione da protocolli come CoAP, OMA LWM2M, OPC, Bluetooth o ZigBee.

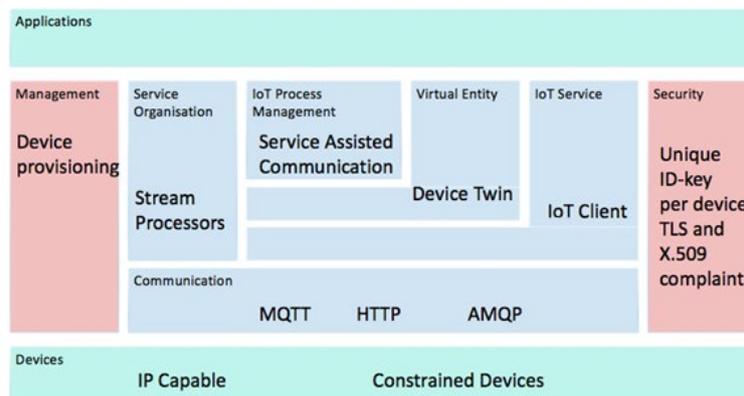


FIGURA 4 MICROSOFT AZURE

## 2.6 I DISPOSITIVI INDOSSABILI

I fitness tracker indossabili come Fitbit o Xiaomi hanno invaso il mercato dei consumatori e reso onnipresente la raccolta di dati personali sulla salute e il fitness. Questi dispositivi stanno diventando, anzi lo sono già, parte integrante della vita quotidiana. Grazie a questi dispositivi un soggetto che li utilizza può migliorare il proprio stile di vita, oppure salvare la propria vita, in quanto grazie alla raccolta dei dati, si possono assicurare interventi tempestivi in caso di situazioni di emergenza. I dispositivi dispongono di diversi tipi di sensori, accelerometro, cardio frequenzimetro, rilevatore della temperatura cutanea, inoltre dispongono di funzionalità aggiuntive, quali le calorie bruciate, stima della distanza percorsa e così via.

Oggi i costi di questi dispositivi non sono così proibitivi, e la loro capacità di raccogliere dati biometrici in tempo reale li rende un ottimo strumento per gestire, se non addirittura prevenire, alcune malattie croniche [1].

Alcuni studi affermano che il numero di applicazione riguardanti il fitness e la salute, che vengono eseguite sugli smartphone, sono destinate a salire, così come l'aumento di dispositivi di monitoraggio.

Tuttavia ancora non è chiaro come i dati generati da questa moltitudine di sensori indossabili possano essere integrate con delle vere e proprie cartelle cliniche. Ogni applicazione scarica i dati biometrici nella propria piattaforma. E' possibile accedere ad alcuni di questi dati attraverso delle specifiche API, ma non sempre è possibile una integrazione totale con altre piattaforme, anche per i vincoli dettati dalle leggi che governano la privacy di dati sensibili. Nel progetto europeo denominato INTER-IoT[2] si è cercato di risolvere problematiche di integrazione offrendo soluzioni software, e poi attraverso una sperimentazione sul campo: i dati biometrici raccolti venivano inviati ad una piattaforma, attraverso la quale, i medici potevano consultarli e renderli parte integrante della cartella clinica di uno specifico soggetto. In questo modo i medici possono avere un quadro completo di un soggetto, che è il risultato della combinazione di dati provenienti da dispositivi indossabili e dalle cartelle cliniche ufficiali, e pertanto potranno proporre consulenze e cure migliori.

Attualmente i dati raccolti dai dispositivi, delle migliori case produttrici oggi in commercio, sono confinati sulle rispettive piattaforme proprietarie. Ogni utente utilizza in maniera autonoma queste applicazioni, definendo i propri obiettivi e cambiando il proprio stile di vita senza una supervisione. Tuttavia una serie di fornitori di servizi sanitari sta pensando alla potenzialità degli smartphone e dei dispositivi indossabili per fornire servizi sanitari supervisionati (interventi a domicilio, assistenza agli anziani).

Per poter far fronte a questo i proprietari di piattaforme hanno rilasciato dei toolkit software, il cui compito è quello di dare allo sviluppatore tutti gli strumenti necessari per accelerare lo sviluppo di applicazioni che possano far fronte alle necessità che si verranno a creare in ambito fitness o sanitario. In questo caso le applicazioni dovranno essere in grado di misurare, visualizzare e inviare dei dati.

I dati che vengono memorizzati sono dati sensibili, quindi bisogna investire in software e hardware affinché i dati siano archiviati e mantenuti in sicurezza, soprattutto in ambito sanitario.

## 2.7 STATO DELL'ARTE

Attualmente esistono tre tipologie di sensori indossabili e che sono commerciabili, suddivise a loro volta in sotto tipologie come di seguito elencate:

- **Accessories**
- **Wrist-worm**
  - Smartwatch
  - Wrist bands
- **Head-mounted**
  - Smart eyewear
  - Headsets/Ear-buds (cuffie e auricolari)
- **Others**
  - Smart Jewellery
  - Straps
- **E-Textiles**
  - Smart garments
  - Foot/ Hand-worm
- **E-Patches**
  - Sensor patch
  - E-tattoo/E-skin

**Accessories:** vengono definiti quei dispositivi indossabili indossati esternamente dagli utenti e non rientrano nella categoria dei principali capi di abbigliamento. Gli accessori fanno parte della categoria di dispositivi più popolare dell'attuale mercato indossabile. Tali dispositivi includono smartwatch, cinturini da polso, occhiali intelligenti, cinturini pettorali, cinture intelligenti e vari clip per abbigliamento. Di seguito vengono descritti i prodotti esistenti in Accessories in tre sottocategorie insieme ai loro casi d'uso.

**Wrist-worm:** come suggerisce il nome, sono i dispositivi indossabili che vengono indossati al polso. I prodotti commerciali sono gli **smart-watch** e i **wrist-band**.

- **Smart-watch:** la funzionalità degli attuali smart-watch è duplice. In primo luogo, fungono da strumenti di comunicazione e notifica integrando gli smartphone con funzionalità come la ricezione di notifiche (ad es. Chiamate telefoniche, SMS, e-mail, controllo vocale e aggiornamenti meteo) ed eseguono micro interazioni (ad es. Avviare un'app nel telefono, limitato Navigazione Web, aggiunta di promemoria e comandi vocali). In secondo luogo, la maggior parte degli smartwatch può anche monitorare alcuni segnali fisiologici umani e quindi agire come dispositivi di tracciamento delle loro attività quotidiane come la registrazione automatica dei tempi di allenamento, il monitoraggio della frequenza cardiaca, il conteggio dei passi e le calorie bruciate. I dati raccolti vengono trasferiti sullo smartphone o su un server cloud per analisi e una migliore presentazione per gli utenti, ad esempio tramite dashboard. Un esempio sono Moto 360, Apple iWatch, ecc.
- **Wrist-band:** sono l'altra categoria popolare di dispositivi indossabili da polso. Mentre ci sono somiglianze con gli orologi intelligenti, i wrist-band sono progettati appositamente per tenere traccia di una serie focalizzata di attività di salute e benessere. La maggior parte dei wrist-band non ha uno schermo per le notifiche o presenta un fattore di forma limitato rispetto agli orologi intelligenti che mirano a sostituire gli orologi convenzionali. Esempi di wrist-band sono i Xiaomi MiBand o UP4 Jawbone

Le due principali sottocategorie di orologi intelligenti e cinturini da polso al momento rispondono a due diverse esigenze degli utenti; sostituiscono l'orologio da polso tradizionale e possono essere considerati come una estensione dello smartphone utilizzati per il rilevamento le attività di fitness. È probabile che questi due tipi di prodotti si fondano in futuro oppure continueranno ad esistere entrambi, e ci saranno wrist-band più sofisticati per gli utenti che necessitano di analisi avanzate nel campo del monitoraggio dell'attività fisica.

**Head-mounted devices:** rappresentano quei dispositivi che si indossano in testa. Esistono due principali sottocategorie, ovvero:

- **Smart Eyewear:** includono occhiali o lenti a contatto con rilevamento, comunicazione wireless o altre funzionalità come la realtà virtuale o aumentata. Un esempio sono le Microsoft HoloLens, Figura 11.
- **Cuffie e auricolari:** stanno diventando dispositivi indossabili con funzionalità più intelligenti e maggiore attenzione al monitoraggio fitness. I vantaggi di tenere traccia delle attività di fitness utilizzando un auricolare sono duplici. Innanzitutto, l'accuratezza può essere maggiore per alcuni tipi di misurazioni come la frequenza cardiaca. In secondo luogo, consente all'utente di ricevere feedback non invasivi in tempo reale come voce che può essere utile nelle attività di fitness. Di conseguenza, in futuro gli hearable potranno offrire una forte concorrenza ai cinturini da polso in quanto possono fornire funzioni intelligenti artificiali avanzate come assistenza vocale e feedback in tempo reale. Tuttavia, essendo dispositivi più piccoli potrebbero non essere in grado di supportare la potenza e le capacità di elaborazione che i cinturini da polso possono offrire.



FIGURA 5 MICROSOFT HOLOLENS

**E-Textiles:** sono una categoria emergente di dispositivi indossabili con applicazioni in molte discipline, come l'assistenza sanitaria, il fitness, gli sport di precisione, la salute e la sicurezza. Esistono numerose attività di ricerca nel campo della scienza dei materiali, nello sviluppo di tessuti conduttivi e nell'inclusione di sensori nei tessuti. Possono essere classificati come

- **Indumenti intelligenti:** Google Project Jacquard (<https://atap.google.com/jacquard/>) ne è un esempio
- **Dispositivi indossati a piede/mano:** Goldfinger è un prototipo di guanto di interfaccia uomo-macchina. Utilizza trasduttori piezoelettrici collegati alle dita per convertire l'energia meccanica generata dalle dita in elettricità per caricare la batteria dei guanti, risparmiando la necessità di cavi e alimentatori voluminosi. L'idea è di usare il guanto per azionare e controllare le macchine con gesti e gli autori immaginano applicazioni nella realtà virtuale e nelle industrie mediche.

**E-Patch:** sono la categoria più recente di dispositivi indossabili che non sono ancora ben consolidati come prodotti di consumo. Simile ai prodotti esistenti, li classifichiamo come **Sensor Patch** e ii) **E-Tattoo / E-Skin**.

- **Sensor Patch:** Alcune ricerche hanno portato allo sviluppo di una patch adesiva che monitora i segnali ECG ed EEG. Il dispositivo può essere caricato in modalità wireless e trasmettere dati a un computer in tempo reale. Questo sensore è ben accoppiato alla pelle e consente ai pazienti di muoversi e comportarsi in modo naturale durante il periodo di monitoraggio. Un altro dispositivo è in grado di rilevare le sollecitazioni e la tensione sulla punta del dito e valutare le proprietà elettriche dei tessuti per aiutare i medici a rimuovere i tessuti con precisione mediante ablazione locale. Questa applicazione può essere potenzialmente utilizzata per creare guanti chirurgici elettronici.

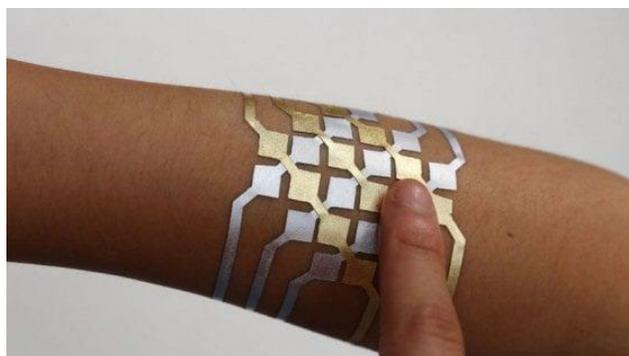


FIGURA 6 E-SKIN

- **E-tattoo/E-Skin:** DuoSkin<sup>2</sup> è un processo di fabbricazione di interfacce utente sulla pelle. La foglia d'oro è stata scelta come materiale chiave per la sua accessibilità, elasticità e comodità quando è attaccata alla pelle. In questo processo di fabbricazione è possibile ottenere tre funzionalità, ovvero il rilevamento dell'input tattile, il display termocromico e la comunicazione wireless con altri dispositivi tramite NFC.

**Altri accessori:** riguardano accessori progettati per essere indossati in altre aree del corpo umano. Si possono classificare in a)

- **Smart Jewellery:** comprende prodotti come anelli intelligenti, collane e clip. Poiché sono di piccole dimensioni, questi dispositivi tendono ad avere caratteristiche specifiche, mirate e limitate rispetto alle funzioni per uso generale fornite da orologi intelligenti o cinturini da polso. I gioielli tipicamente intelligenti sono progettati per avvisare gli utenti in merito alle notifiche degli smartphone, effettuare pagamenti, rintracciare segnali fisiologici umani o rilevamento ambientale. Poiché i prodotti di questi tipi sono meno comuni e abbastanza nuovi per il mercato dei dispositivi indossabili, le informazioni tecniche disponibili e le recensioni sono limitate.
- **Straps:** cinturini pettorali, cinture, fasce da braccio (fissate sulla parte superiore del braccio in contrasto con un cinturino da polso) o cinghie per ginocchia dotate di sensori per il monitoraggio della salute o altre funzionalità sono classificate come cinturini intelligenti. Le cinghie intelligenti trovano applicazioni nel monitoraggio dei segnali fisiologici umani e della biomeccanica e nelle applicazioni di rilevamento e tattili.

### 3 HEALTH/FITNESS KIT CONNECTOR

#### 3.1 GOOGLE FIT

Google Fit [1][3] è un sistema aperto che permette a tutti gli sviluppatori di applicazioni di memorizzare i dati, provenienti da sensori indossabili, in un repository centrale, a cui gli utenti possono accedere con qualsiasi applicazione e qualsiasi dispositivo. Le applicazioni saranno in grado di interfacciarsi con qualsiasi sensore, o altro dispositivo, recuperando i dati e facendo in modo che questi rimangano inalterati anche quando i dispositivi indossabili vengono aggiornati.

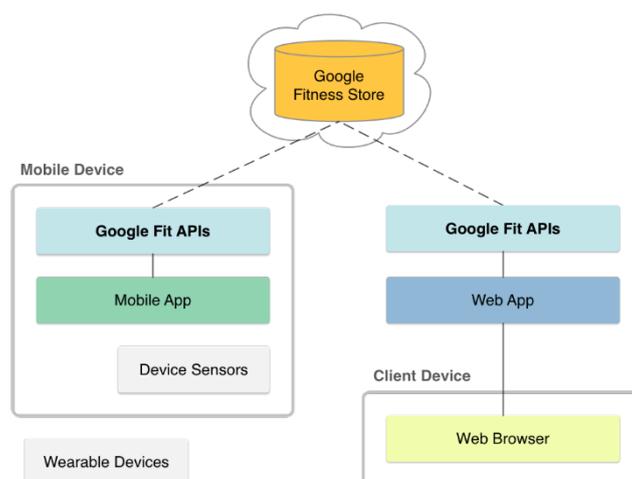


FIGURA 7 GOOGLE FIT

<sup>2</sup> <https://duoskin.media.mit.edu/>

Google Fit fa molta attenzione ai dati che tratta, infatti informa i propri sviluppatori quali sono le condizioni per il trattamento dei dati: ad esempio non è ancora possibile memorizzare dati biometrici e medici. Bisogna sempre informare l'utente sullo scopo della memorizzazione dei dati.

I componenti della piattaforma Google Fit sono i seguenti:

### 3.2 THE FITNESS STORE

Rappresenta il repository centrale, attraverso il quale è possibile memorizzare una varietà di dati provenienti da diverse tipologie di sensori indossabili. Si tratta di un servizio di cloud che utilizza l'infrastruttura di Google per memorizzare i dati. Fornisce delle API attraverso le quali le applicazioni possono accedere ai dati creati da altre applicazioni.

### 3.3 THE SENSOR FRAMEWORK

Definisce un insieme di alto livello per i sensori, i tipi di dati, le sessioni. Ha lo scopo di facilitare la memorizzazione dei dati.

### 3.4 PERMISSIONS AND USER CONTROLS

Definisce un insieme di regole e permessi per poter manipolare i dati di fitness.

### 3.5 GOOGLE FIT APIS

API Android e REST sono utilizzati per accedere al fitness store. Si possono creare applicazioni che supportano Google Fit su più piattaforme e dispositivi, come Android, iOS e app Web.

### 3.6 SAMSUNG DIGITAL HEALTH PLATFORM (SDH)

La Figura 8 illustra il modello di implementazione della piattaforma Samsung Digital Health (SDH)[1]. Gli oggetti principali sono l'applicazione Samsung Health (S Health) e i server cloud SDH. Simile a Health Store di Apple, gli sviluppatori di applicazioni possono utilizzare S Health per archiviare e accedere ai loro dati. SDH mira a svolgere un ruolo attivo anche a livello di servizio. I dati archiviati in S Health sono sincronizzati con i server SDH, che rappresentano l'architettura di servizio per Samsung Health Platform ed è possibile accedervi, in sicurezza, direttamente da altri fornitori di servizi utilizzando delle specifiche API. La piattaforma è open source lato dispositivo, poiché il sistema su cui le applicazioni vengono eseguite è Android. Samsung è anche impegnata nello sviluppo di SIMBAND ([www.simband.io](http://www.simband.io)), un dispositivo sanitario generico. Sia i dispositivi di salute e fitness basati su Android Wear, sia quelli basati su SIMBAND possono connettersi a SDH. SDH utilizza un modello di dati simile a quello di Apple. Inoltre tale modello è personalizzabile, cioè può essere esteso con un proprio modello, dando piena libertà agli sviluppatori.

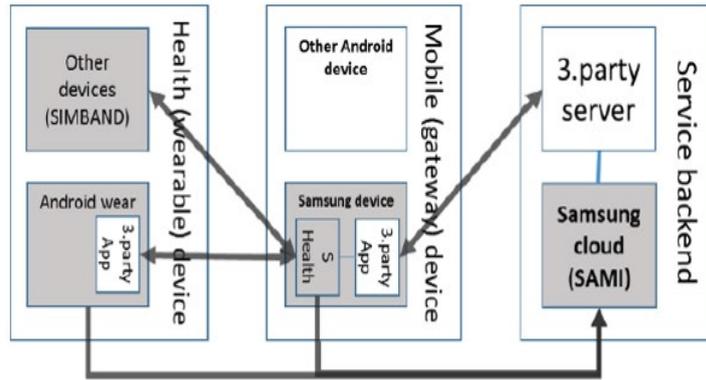


FIGURA SEQ FIGURA \\* ARABIC 8  
IMPLEMENTAZIONE S HEALTH

### 3.7 APPLE HEALTHKIT

Le principali risorse di Apple HealthKit [1] sono rappresentate dall' Health Store (HS) e dalle API per accedere al contenuto di HS come si può notare in Figura 9. HS rappresenta l'archivio dati per tutte le applicazioni di salute e fitness sviluppate per iOS. Le applicazioni di salute sviluppate da Apple e terze parti possono archiviare i dati in HS e condividerli con altre applicazioni sullo stesso dispositivo. Attraverso meccanismi rigorosi di sicurezza i dati sono accessibili solo localmente. Tuttavia, il contenuto di HS può essere facoltativamente esportato sui server iCloud di Apple sotto forma di un file XML crittografato a scopo di backup. I dati che vengono archiviati in HS possono essere estesi, dando la possibilità agli sviluppatori di terze parti di creare un proprio set di dati, e allo stesso tempo condividerli. HealthKit, come altri prodotti Apple, può essere eseguito su dispositivi iOS, e può funzionare solo con un numero limitato di dispositivi Bluetooth certificati di terze parti. Inoltre Apple ha cercato di sviluppare partnership con i fornitori di servizi sanitari al fine di promuovere HealthKit come front-end del servizio sanitario. Secondo quanto riferito, Apple ha avviato una sperimentazione clinica in collaborazione con Epic Systems e Mayo.

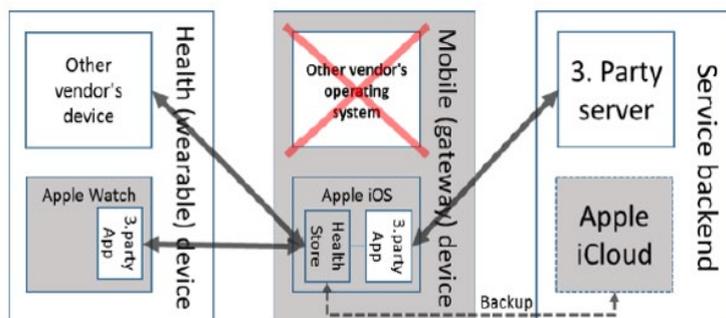


FIGURA 9 HEALTH STORE APPLE

### 3.8 OPEN-MHEALTH

Open mhealth [4] si prefigge come obiettivo principale la standardizzazione dei dati: dare la possibilità a tutta la comunità open source di poter accedere alle informazioni generate dal paziente e allo stesso tempo renderli comprensibili. Inoltre si prefigge la memorizzazione ed integrazione di flussi eterogenei di dati (vedi architettura Figura 10).

#### 3.8.1 STANDARDIZZARE I DATI.

La piattaforma si basa su dati sanitari strutturati, che aiutano aziende, organizzazioni e individui a scambiarsi dati e riutilizzare il codice. Inoltre semplifica la comprensione degli stessi, utilizzando schemi per definire la struttura dei dati sanitari, e avvalendosi di esperti clinici, data scientist, sviluppatori e architetti di software per elaborare schemi semplici, estensibili e clinicamente validi per i tipi di dati più comuni e importanti in ambito sanitario.

#### 3.8.2 MEMORIZZAZIONE DEI DATI

La memorizzazione dei dati avviene attraverso delle API RESTful che lo sviluppatore può utilizzare in maniera trasparente senza sapere da e dove i dati vengono memorizzati. L'API utilizza e produce dati conformi agli schemi e autorizza l'accesso tramite OAuth 2.0. È supportato da MongoDB ed è facile da implementare tramite Docker.

#### 3.8.3 INTEGRAZIONE DI FLUSSI

L'integrazione dei flussi avviene attraverso degli adattatori open-source che raccolgono dati sanitari utilizzando le API di grandi fornitori come RunKeeper, Fitbit, Google e Apple, e poi vengono convertiti per essere adattati agli schemi mhealth. Questi dati possono quindi essere elaborati o visualizzati indipendentemente da dove provengono, consentendo di creare velocemente prototipi o creare applicazioni pronte per la produzione.

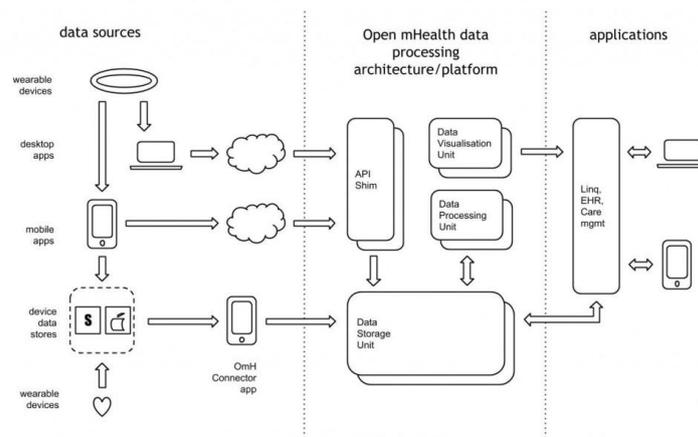


FIGURA 10 OPEN-MHEALTH

### 3.9 COMMUNITY HEALTH TOOLKIT

Community Health Toolkit (CHT) [5] è un progetto di un gruppo di organizzazioni leader che si sono riunite per sostenere lo sviluppo di iniziative di salute digitale nelle aree più difficili da raggiungere. Fornisce una raccolta di tecnologie open source e risorse di progettazione, tecnica e di implementazione open source che consentono di creare e distribuire strumenti digitali per la salute della comunità. L'obiettivo è un modello sanitario equo, universale, di qualità, accessibile a tutti. Le tecnologie usate sono rappresentate in Figura 5.

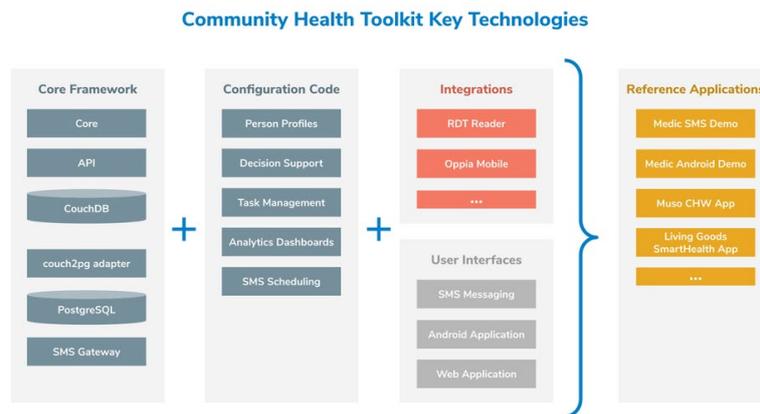


FIGURA 11 COMMUNITY HEALTH TOOLKIT TECHNOLOGIES

Community Health Application Framework supporta un'ampia personalizzazione e rende il processo di sviluppo più veloce, più affidabile, più interoperabile e più sicuro rispetto alla codifica da zero di un'app di salute digitale. Il Core Framework consente agli sviluppatori di implementare rapidamente più funzionalità e renderle il più possibile personalizzabili. La facilità con cui è possibile sviluppare una applicazione, abbinata alla flessibilità, ha reso CHT particolarmente utile per le applicazioni di salute della comunità, che spesso hanno complesse esigenze di gestione delle attività, supporto decisionale e gestione delle prestazioni.

La personalizzazione di un'applicazione avviene usando linguaggi come XForms, JSON e JavaScript che configurano le funzionalità del framework per essere eseguito in un modo specifico. Ogni elemento dell'applicazione viene definito in modo modulare, quindi si specifica quando e come dovrebbe apparire nell'interfaccia utente. In questo modo, è possibile scrivere codice che personalizza la modalità di visualizzazione delle funzionalità principali per diversi tipi di utenti, senza dover modificare il framework sottostante. Collettivamente, questa personalizzazione viene definita codice di configurazione.

La progettazione di un'applicazione per la salute della comunità che si integri con l'ecosistema di salute digitale più ampio è una potente opportunità per supportare l'assistenza ai pazienti in maniera più integrata e proattiva. CHT è progettato per integrare le applicazioni autonome in esecuzione sullo smartphone o sul tablet degli operatori sanitari e per supportare integrazioni di backend più complesse tramite le API del core del framework.

### 3.10 MONITORAGGIO CON NODO-EDGE

Le tecnologie dell'informazione e della comunicazione stanno trasformando le interazioni sociali, lo stile di vita e i luoghi di lavoro. Una delle applicazioni più promettenti della tecnologia dell'informazione è la gestione della salute e del benessere. L'assistenza sanitaria si sta spostando dalle risposte reattive alle condizioni acute a un approccio proattivo caratterizzato da diagnosi precoce, prevenzione e gestione dell'assistenza sanitaria a lungo termine. In questo quadro, il monitoraggio delle condizioni di salute e la gestione del benessere sono

considerati come un contributo significativo all'assistenza sanitaria e al benessere individuali. Ciò è particolarmente importante nei paesi sviluppati con una significativa popolazione che invecchia, dove la tecnologia dell'informazione può essere impiegata per migliorare significativamente la gestione delle condizioni croniche e, quindi, la qualità della vita complessiva. La registrazione continua o persino occasionale di segnali biomedici è particolarmente critica per la diagnosi e trattamento delle malattie cardiovascolari. Ad esempio, la registrazione continua di un elettrocardiogramma (ECG) o fotoplethysmogram (PPG) da un sensore indossabile fornisce una visione realistica delle condizioni cardiache di un paziente monitorando fattori quali ipertensione, stress, ansia, diabete e depressione, durante le normali routine quotidiane. Inoltre, l'analisi automatizzata di tali segnali biomedici registrati supporta i medici nel loro lavoro quotidiano e consente lo sviluppo di sistemi di allarme. Ciò comporta numerosi vantaggi, come la riduzione dei costi sanitari, aumentando l'osservabilità della salute, la collaborazione tra medici e l'efficienza medico-paziente. Inoltre, il monitoraggio continuo serve ad aumentare la diagnosi precoce di condizioni di salute e malattie anormali, offrendo un modo per migliorare la qualità della vita dei pazienti. Oggi, una maggiore attenzione è focalizzata sulla prevenzione e la diagnosi precoce delle malattie, nonché sulla gestione ottimale delle condizioni croniche. Queste funzioni sono spesso potenziate da nuove tecnologie indipendenti dalla posizione. Per realizzare appieno un ambiente pervasivo o onnipresente, le reti di area personale (PAN) devono essere connesse a reti basate su protocollo Internet (IP). Tale integrazione consente la condivisione delle risorse all'interno delle reti, massimizzando l'utilizzo delle risorse disponibili. Inoltre, la comunicazione con i singoli nodi in una rete richiede un meccanismo di indirizzamento efficiente.

Inoltre, i telefoni cellulari di nuova generazione hanno un impatto importante sullo sviluppo di tali sistemi sanitari, in quanto integrano perfettamente una vasta gamma di reti (3G, Bluetooth, LAN wireless e GSM) attraverso punti di accesso (AP), offrendo così l'opportunità di trasmettere segnali biomedici registrati a un server centrale in un ospedale. Di conseguenza, il monitoraggio continuo dei segnali biomedici non sarà più limitato all'ambiente domestico.

I dispositivi di comunicazione mobile possono ora fornire servizi efficienti e convenienti, come lo scambio di informazioni in remoto e l'accesso alle risorse tramite dispositivi mobili, consentendo agli utenti di lavorare in modo onnipresente. Con la crescita astronomica del tasso di proprietà dei telefoni cellulari, l'assistenza sanitaria mobile supportata da tecnologie mobili e wireless emerge come una soluzione di assistenza economica con un risultato sanitario complessivo migliore. Un dispositivo mobile fattibile per l'assistenza sanitaria onnipresente deve essere economico da produrre, ultracompatto, leggero e il suo consumo energetico deve essere basso. Oltre alle ampie capacità di comunicazione, deve supportare tali funzioni di monitoraggio delle condizioni di salute e visualizzazione di segnali biomedici. È ora possibile trarre inferenze in tempo reale da una serie di dati comportamentali resi disponibili tramite telefoni cellulari. È quindi possibile fornire feedback in merito a questi comportamenti, consentendo alle persone di fare scelte di vita quotidiane migliori e, in definitiva, di gestire meglio la propria salute.

Tuttavia, sebbene si preveda che nuovi servizi sanitari riducano i costi e aumentino la qualità della vita degli utenti, maggiore è il numero di organizzazioni di dati che raccolgono, da dispositivi di monitoraggio medici fissi e mobili, tanto più difficile è l'uso efficace di un'elaborazione centralizzata dei dati basata su cloud e repository [10]. In effetti, nel caso di applicazioni quasi in tempo reale, l'uso massiccio dell'infrastruttura basata su cloud potrebbe influire sui vincoli in tempo reale e gravare sull'infrastruttura di rete dal locale al cloud. Inoltre, per fornire servizi sanitari a una larga scala di pazienti, approcci semplicistici, in cui l'infrastruttura tra i dispositivi di rilevamento e il cloud vengono utilizzati solo come infrastruttura di comunicazione comune, non sono spesso fattibili a causa della presenza di altre sfide sanitarie. Ad esempio, in alcuni casi, per garantire la privacy del paziente, i dati non possono essere archiviati nel cloud pubblico o, in altri casi, per la sicurezza del paziente, i dati devono essere immediatamente disponibili e eventuali ritardi o guasti introdotti dal cloud non possono essere tollerati. La velocità dei dati e delle analisi è essenziale in molte applicazioni IoT industriali, ma è anche un elemento chiave dell'assistenza sanitaria industriale e di tutte le altre aree in cui ci si muove verso decisioni autonome e semi-autonome prese da sistemi, attuatori e vari controlli. In particolare, il grado di autonomia è uno degli obiettivi più importanti e desiderati della prossima fase di sviluppo di Industry 4.0. Pertanto, per una serie di ragioni (larghezza di banda, costi, velocità,

analitica predittiva, sicurezza, manutenzione) è necessario implementare un approccio più rapido, economico e intelligente rispetto a quello tradizionale basato su cloud.

Un potenziale scenario che tenga conto di quanto sopra è stato detto può essere quello rappresentato in Figura 12

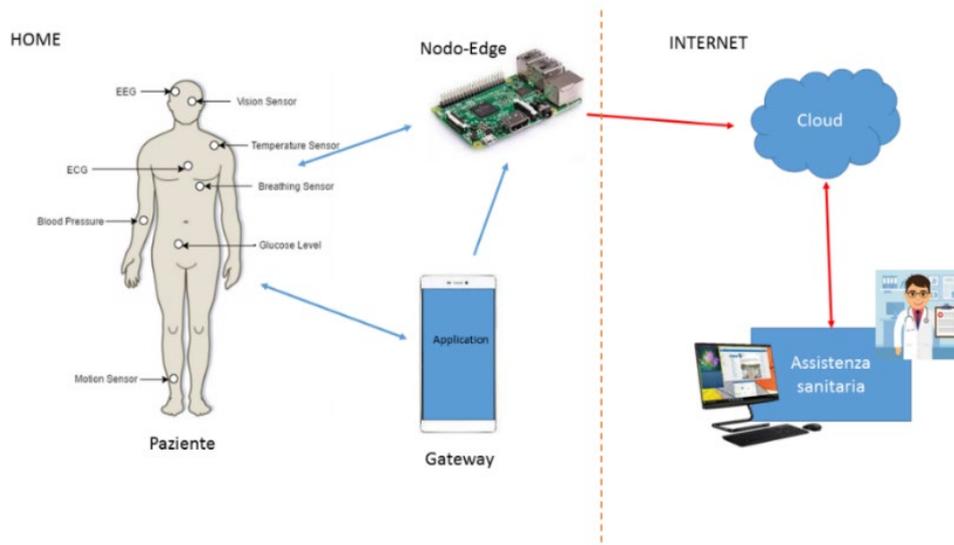


FIGURA 12 MONITORAGGIO CON NODO-EDGE

Un paziente viene monitorato costantemente con dei sensori indossabili. I dati sono raccolti dal nodo edge che in questo caso è rappresentato da un Raspberry pi. Inoltre per alcune categorie di sensori indossabili è necessario usare un'applicazione che viene eseguita su smartphone per interfacciarsi e recuperare le misure di questi sensori. Il nodo-edge ha il compito di elaborare in maniera veloce i dati ricevuti, ed inoltrare verso il cloud le relative diagnosi. Attraverso il cloud i medici possono ricevere tali diagnosi per un'ulteriore esame ed, eventualmente, valutare un pronto intervento che potrebbe salvare la vita al paziente. E' stato proposto un sistema per il monitoraggio della frequenza cardiaca, della temperatura corporea della persona con sensori dedicati insieme ad un Raspberry pi. I dati vengono raccolti sul cloud Bluemix, utilizzando il protocollo MQTT(Message Queuing Telemetry Transport) e possono essere recuperati dal medico per essere analizzati ovunque e qualsiasi criticità verrà rilevata tempestivamente. Un perfetto compromesso tra accuratezza e costo del sistema viene realizzato scegliendo i sensori appropriati che sono DS18B20 (sensore di temperatura) e KG011 (sensore di frequenza cardiaca). La frequenza del polso e la temperatura di una persona in momenti diversi vengono misurati dai sensori. Le letture sono mostrate sotto forma di grafici sulla piattaforma IBM Watson IoT.

## 4 NODE-RED: A FLOW-BASED DEVELOPMENT TOOL

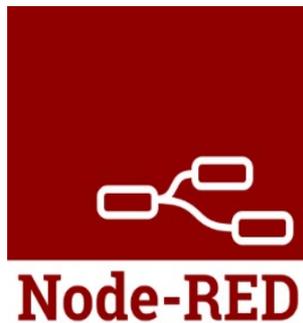


FIGURA 13 – NODE-RED

Node-RED<sup>3</sup> è un potente strumento per la costruzione di applicazioni Internet of Things (IoT) con l'obiettivo di semplificare il "wiring together" dei blocchi di codice per l'esecuzione dei task. Utilizza un approccio di programmazione visuale che permette agli sviluppatori di collegare tra loro blocchi di codice predefiniti, noti come 'nodi', per eseguire un particolare task. I nodi collegati, di solito una combinazione di nodi di ingresso, nodi di elaborazione e nodi di uscita, quando sono cablati insieme, costituiscono un 'flusso'.

Sviluppato originariamente come progetto open source presso IBM alla fine del 2013, per soddisfare la loro necessità di connettere rapidamente hardware e dispositivi a servizi web e altri software - come una sorta di collante per l'IoT - si è rapidamente evoluto fino a diventare uno strumento di programmazione generale per l'IoT. È importante sottolineare che Node-RED ha rapidamente sviluppato una significativa e crescente base di utenti e una comunità di sviluppatori attivi che stanno contribuendo con nuovi nodi che permettono ai programmatori di riutilizzare il codice Node-RED per un'ampia varietà di compiti.

Una versione iniziale di Node-RED è stata rilasciata come progetto open source alla fine del 2013 e ha creato un piccolo ma attivo gruppo di utenti e sviluppatori nel corso del 2014. Al momento della scrittura, Node-RED è ancora una tecnologia emergente, ma ha già visto un'adozione significativa da parte di makers, sperimentatori e un certo numero di grandi e piccole aziende che stanno sperimentando il suo utilizzo per le loro esigenze. Oggi c'è una vivace comunità di utenti e sviluppatori, con un gruppo centrale che lavora sul codice Node-RED stesso e la maggior parte degli sviluppatori che contribuiscono con nodi o flussi alla libreria di flusso.

---

<sup>3</sup> <https://nodered.org/>

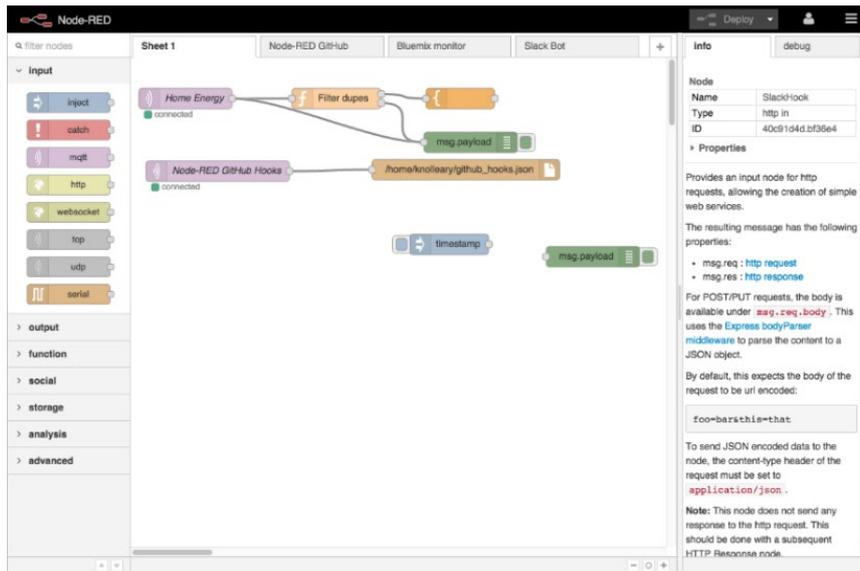


FIGURA 13 – ESEMPIO INTERFACCIA NODE-RED

Sebbene Node-RED sia stato originariamente progettato per funzionare con l'Internet of Things, cioè dispositivi che interagiscono e controllano il mondo reale, man mano che si è evoluto, è diventato utile per una serie di applicazioni.

#### 4.1 I FLUSSI

Un primo flusso di esempio, riportato nella figura seguente, riceve un messaggio da un feed di Twitter contenente un hashtag e accende/spegne un LED quando arriverà il nuovo tweet. Per questo esempio si suppone che NODE-RED sia in esecuzione su un Raspberry PI- una situazione comune per molti utenti di Node-RED.

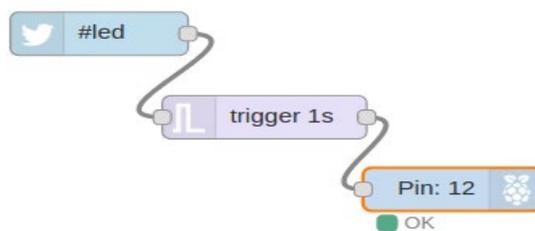


FIGURA 14 UN SEMPLICE FLUSSO A 3 NODI CHE UTILIZZA UN TAG TWITTER PER ATTIVARE UN LED SU UN RASPBERRY PI.

Il flusso Node-RED rileva un hashtag #led ed accende un LED collegato ad un Raspberry PI. Il sistema è composto da 3 nodi collegati tra loro - un nodo twitter a sinistra, un nodo trigger al centro e un nodo Raspberry PI (gpionode) a destra (gpio sta per General Purpose Input/Output ed è il nome dato al generico

nodo Raspberry PI che gestisce IO). I nodi, rappresentati da caselle arrotondate, hanno di solito un'icona per rappresentare il loro tipo, ad esempio l'icona di Twitter nel primo nodo, e un nome di testo che può essere impostato dal programmatore.

Ognuno di questi tre nodi è integrato nello strumento di programmazione visiva Node-RED e può essere trascinato da una tavolozza di nodi nell'area di lavoro principale. Essi vengono poi 'cablati' insieme unendo le loro schede di uscita alla scheda di ingresso del nodo successivo nel flusso (le piccole macchie grigie a destra o a sinistra delle caselle del nodo.)

Il nodo Twitter è un nodo integrato in Node-RED e nasconde tutta la complessità dell'utilizzo delle API di Twitter. Può essere configurato con le credenziali dell'account di un utente e un set di stringhe di ricerca, in questo caso si tratta semplicemente di cercare l'hashtag '#led'. Quando il nodo Twitter rileva il tag in un tweet all'utente, o nel feed del tweet pubblico, crea un nuovo messaggio con i dettagli del tweet, che viene inoltrato al nodo successivo del flusso. Nel nostro esempio, poiché l'output del nodo Twitter è collegato all'input del nodo trigger, il nodo trigger riceve un messaggio con i dettagli del tweet.

Il nodo trigger è un altro nodo integrato in Node-RED e il suo comportamento predefinito è quello di attendere qualsiasi messaggio sul suo input. Quando riceve un messaggio, si 'innesca', e invia un messaggio in uscita con il valore "1" come corpo del messaggio. Quindi attende 1 secondo e invia un secondo messaggio con il valore "0" nel corpo del messaggio. Poiché il nodo trigger è collegato al gpionode, che controlla i pin Input/Output o IO sul Raspberry PI, riceve questi due messaggi, a distanza di 1 secondo l'uno dall'altro, sul suo ingresso.

Anche il gpionode è un nodo built-in (pre-incorporato) in Node-RED che nasconde i dettagli dell'impostazione dei pin di IO sul PI. Quando riceve un messaggio con un valore di "1" nel corpo, prende il pin IO in alto, e quando riceve un messaggio con corpo "0", prende il pin in basso. In questo esempio, il gpionode è configurato per controllare il pin numero 12, e quindi il nodo lo mostra nella relativa icona del nodo.

Se il Raspberry PI è cablato con un LED collegato al pin 12, il gpionode che va in alto per 1 secondo e poi in basso, farà sì che il LED lampeggi per 1 secondo. L' esempio fa comprendere con sia incredibilmente potente linsieme di Node-RED e dell'Internet of Things (IoT). Collegare un Raspberry PI a un LED, non richiede alcuna programmazione - tutti i compiti sono svolti da nodi pre-costruiti in Node-RED che hanno solo bisogno di essere configurati. Questo dimostra davvero la potenza di Node-RED quando si esegue su un dispositivo semplice come un PI Raspberry.

Un secondo esempio mostra un diverso utilizzo di Node-RED. Piuttosto che controllare un dispositivo come il Raspberry PI, questo flusso combina le informazioni meteorologiche e il vostro utilizzo del fitbit per inviarvi un'e-mail di avviso quando il tempo è buono e siete in ritardo sul vostro programma di allenamento. Il flusso risulta ovviamente più complesso, ma anche in questo caso, utilizzando i nodi built-in si possono eseguire task piuttosto complessi con poca o nessuna programmazione.

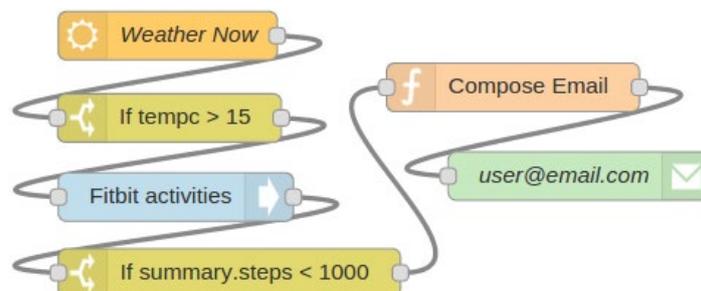


FIGURA 15 FLUSSO PER IL SERVIZIO DI ALERT METEO

Come nell'esempio precedente, i nodi vengono trascinati dalla tavolozza dell'interfaccia utente Node-RED e collegati tra loro. Partendo dal nodo più a sinistra, si ha un nodo `openweather` che è possibile configurare per controllare il meteo di qualsiasi grande città a intervalli prestabiliti. Una volta recuperati, i risultati vengono incapsulati in un messaggio che viene inoltrato al nodo successivo, in questo caso, ad un nodo di commutazione. Il nodo `switch` esegue un semplice test, controllando il messaggio in ingresso soddisfa alcuni requisiti e inviando un messaggio con un valore "1" se il test è vero. In questo caso, è stato configurato per testare il valore della temperatura (`tempc`) per vedere se è maggiore o uguale a 15 gradi centigradi.

Quando il test del nodo di commutazione restituisce `true`, o "si accende", invia un messaggio sul nodo `fitbit` che è il nodo successivo nel flusso. Il nodo `fitbit` è un altro potente nodo integrato che è possibile configurare per ottenere statistiche sul proprio dispositivo `fitbit` dal portale `fitbit.com`. Il `fitbit` node impacchetta le statistiche del vostro dispositivo in un messaggio e le inoltra al nodo successivo nel flusso.

Il nodo successivo è un altro nodo `switch`. Questo è stato configurato per verificare se il numero di passi fatti oggi è inferiore a 1000. Viene testato il campo dati `fitbit.steps`, `summary.steps`. Queste statistiche sono incapsulate in un messaggio dal nodo `fitbit`, che a sua volta le ha ottenute mediante una chiamata all'API `fitbit` su `fitbit.com`.

Il nodo `funzione` è un generico nodo universale che permette di scrivere la propria logica di programmazione in Javascript. Si può presumere che questo nodo crei solo un nuovo messaggio con un oggetto e un corpo di testo che può essere inviato via e-mail. Una volta che questo messaggio viene creato all'interno del nodo `funzione`, viene inviato attraverso il flusso al nodo finale che è un nodo `email`.

Questo nodo prenderà il corpo di qualsiasi messaggio in arrivo e lo invierà all'account di posta elettronica configurato al momento dell'inizializzazione del nodo.

L'ultimo esempio mostra come creare un semplice servizio web che mediante browser invia una richiesta HTTP interrogando alcuni dati del *mondo reale*, e poi restituisce una pagina web con una rappresentazione grafica di quei dati forniti al browser.

Il flusso è mostrato nella figura successiva e consiste di 4 nodi. Il primo e l'ultimo nodo sono nodi di input e di output `http` che lavorano insieme per ascoltare le richieste HTTP e inviare le risposte HTTP. Questo flusso ascolta una richiesta HTTP da qualsiasi fonte, e quando arriva, interroga le statistiche `fitbit` da `fitbit.com`, poi usa un nodo `template` per costruire una pagina HTTP, e lo passa al nodo di uscita `http` che rimanda la pagina web al browser.

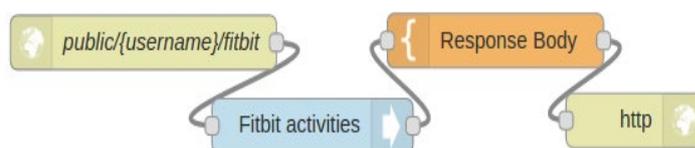


FIGURA 16 UN SEMPLICE ESEMPIO DI WEB SERVER PER GRAFICARE I DATI FITBIT

Il codice HTML nel nodo template è mostrato nel Listato 1.

Listato 1 Template HTML per visualizzare un grafico a ciambella del consumo di calorie

```
1. <!doctype html>
2. <head>
3.   <title>A Node RED Example</title>
4.   <link rel="stylesheet" href="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.css">
5.   <script src="//cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-min.js"></script>
6.   <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
7.   <script src="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.min.js"></script>
8. </head>
9. <html>
10.  <div id="chart-example" style="height: 250px;"></div>
11.  <script>
12.    Morris.Donut({
13.      element: 'chart-example',
14.      data: [
15.        {label: "Activity Calories", value: {{payload.summary.activityCalories}} },
16.        {label: "Calories BMR", value: {{payload.summary.caloriesBMR}} },
17.        {label: "Calories Out", value: {{payload.summary.caloriesOut}} }
18.      ]
19.    });
20.  </script>
21. </html>
```

Una volta che il nodo template ha generato il file HTML, lo passa come messaggio al nodo finale, che è un nodo di risposta http. Questo nodo impacchetta l'HTML come risposta HTTP che invia al browser. L'utente vedrà quindi un semplice grafico a ciambella del suo utilizzo calorico per il giorno - tutto costruito in modo automatico da un semplice flusso Node-RED. Un esempio è riportato nella figura successiva.



FIGURA 17 UN GRAFICO A CHE MOSTRA IL CONTEGGIO DELLE CALORIE DAL FITBIT

## 4.2 NODE-RED E L'INTERNET OF THINGS

Quando gli IBM hanno creato Node-RED, si sono concentrati soprattutto sull'Internet delle cose, cioè sul collegamento dei dispositivi al sistema di elaborazione e viceversa. Come strumento per lo sviluppo rapido di applicazioni per l'IoT, Node-RED è potente e flessibile. La sua potenza deriva da una combinazione di due fattori:

- Node-RED è esempio di un modello di programmazione basato sul flusso - messaggi che rappresentano il flusso di eventi tra i nodi, innescando un'elaborazione che si traduce in output. Il modello di programmazione basato sul flusso mappa bene le tipiche applicazioni dell'Internet degli oggetti che sono caratterizzate da eventi del mondo reale che innescano una sorta di elaborazione che a sua volta si traduce in azioni del mondo reale. Node-RED impacchetta questi eventi come messaggi che offrono un modello semplice ed uniforme per gli eventi che fluiscono tra i nodi che compongono i flussi.
- L'insieme dei nodi integrati Node-RED rappresentano un insieme di potenti nodi di input e di output, ognuno dei quali nasconde gran parte della complessità dell'interazione con il mondo reale. Node-RED offre agli sviluppatori potenti blocchi di costruzione per consentire loro di mettere insieme rapidamente flussi complessi senza doversi preoccupare dei dettagli di programmazione.

Questi due fattori rendono Node-RED un potente strumento per lo sviluppo di applicazioni IoT. Se combinato con la flessibilità di creare e utilizzare nodi come i nodi *funzione*, che permettono allo sviluppatore di scrivere rapidamente JavaScript arbitrario, Node-RED è probabilmente uno degli strumenti principali nel toolbox degli sviluppatori dell'IoT.

Tuttavia, Node-RED non è sempre lo strumento migliore per lo sviluppo di applicazioni IoT. Alcune situazioni in cui Node-RED potrebbe non essere la prima scelta includono:

- Complesse applicazioni multifunzione dell'Internet degli oggetti. Node-RED eccelle nello sviluppo rapido di applicazioni e nell'agire come collante per collegare gli eventi alle azioni, o i sensori agli attuatori, se lo si desidera. Quando un'applicazione supera una certa dimensione, diventa complessa da programmare e gestire visivamente attraverso Node-RED e l'interfaccia utente diventa un collo di bottiglia.
- La programmazione basata sui flussi è un modello di programmazione che non sempre è il più adatto per lo sviluppo di applicazioni. Allo stesso modo in cui alcuni linguaggi di programmazione eccellono in alcuni compiti ma non in altri, la programmazione basata sul flusso ha i suoi punti deboli. Un buon esempio sono i cicli: Node-RED è poco adatto quando si gestiscono i loop.
- Casi d'uso specifici. La programmazione basata sul flusso è un modello di uso generale e non è mirata o ottimizzata per esigenze specifiche, per esempio la Data Analytics o lo sviluppo dell'interfaccia utente. Al momento Node-RED non ha un supporto specifico per questo tipo di applicazioni e non ha un modo semplice per aggiungere tale supporto. Ovviamente, poiché la tecnologia alla base di Node-RED è JavaScript, è possibile sfruttare le sue capacità per queste esigenze.

## 5 CONCLUSIONI

Nel documento è stata descritto il processo di analisi e di valutazione effettuato per realizzare la progettazione e la definizione del framework alla base del sistema K4H, scopo del progetto. L'analisi di dispositivi indossabili porta a comprendere come questi dispositivi fanno sempre più parte del vivere quotidiano e rappresentano uno strumento in più per poter valutare e migliorare lo stile di vita.

Inoltre la possibilità di poter adottare diverse scelte, o valutare diverse soluzioni nel realizzare un valido sistema di monitoraggio a seconda delle proprie esigenze fa sì che possano essere rispettati requisiti di costo e di privacy laddove ce ne sia l'esigenza.

Come si è potuto constatare, le soluzioni possono essere molteplici, dall'utilizzo di piattaforme esistenti, alla loro integrazione, alla possibilità di usare nodi Edge che sono più vicini al soggetto da monitorare, garantendo tempi di reattività minori in caso di particolari criticità.

HealthKit può essere usato solo con dispositivi proprietari (Apple), e quindi le applicazioni di terze parti devono essere basate su iOS. Sebbene i dispositivi iOS siano facili da usare, sono proprietari della sola Apple. Gli HSP dovranno fare affidamento su Apple per espandere l'ecosistema con ad es. nuovi dispositivi per la salute e il fitness di altri fornitori. Google Fit ha un modello incentrato sul cloud, e il suo modello di business è incentrato, principalmente, sulla vendita di pubblicità. Il progettato è focalizzato sulla raccolta ed archiviazione dei dati di fitness dagli utenti di Google che può quindi utilizzare questi dati per pubblicità mirata. Inoltre, Google Fit non pone alcuna limitazione al tipo di dispositivo utilizzato. Anche l'esecuzione di Android non è un requisito. Pertanto gli HSP possono scegliere tra una vasta gamma di dispositivi utente con diversi fattori di forma e funzionalità.

Samsung vuole integrare sistemi HealthKit e Fit, visto che l'azienda si occupa e si è sempre occupata di dispositivi, tra cui SIMBAND, che ha una architettura hardware aperta. Samsung cerca anche di soddisfare le esigenze dei fornitori di servizi, sebbene la sua piattaforma cloud possa affrontare ostacoli in diversi paesi a causa delle normative sulla privacy. Mentre, Open-mhealth e Community health kit sono architetture ancora giovani per competere con questi colossi, ma sono sulla buona strada. Ottima scelta mettere al centro della propria strategia la standardizzazione dei dati, facilitando l'interoperabilità tra piattaforme eterogenee con Open-mhealth.

Per quanto riguarda lo strumento di costruzione delle applicazioni Internet of Things (IoT), la scelta è caduta su Node-RED, con l'obiettivo di semplificare il "wiring together" dei blocchi di codice per l'esecuzione dei task. Node-RED, infatti, utilizza un approccio di programmazione visuale che permette agli sviluppatori di collegare tra loro blocchi di codice predefiniti, noti come 'nodi', per eseguire un particolare task, mediante il cosiddetto "flusso". Mentre per la fase di elaborazione, è stato utilizzato il framework open source Hadoop, che permette di elaborare grandi set di dati, grazie alla capacità di sfruttare la potenza di calcolo e lo spazio di archiviazione di computer facenti parte di cluster, proponendo un modello di programmazione molto semplice. Il framework, infatti, è progettato per scalare da un singolo server a migliaia di macchine, ognuna delle quali mette a disposizione le proprie risorse. L'algoritmo MapReduce utilizzato è una tecnica di elaborazione basata sul calcolo distribuito che contiene due task basilari, Map e Reduce. La funzione Map prende una serie di dati e li converte in un'altra, dove i singoli elementi sono suddivisi in tuple (coppie chiave/valore). Segue la funzione di Reduce, che prende l'output della funzione di Map come input e combina queste tuple di dati in un insieme più piccolo. Un programma MapReduce viene eseguito in tre fasi: Map, Shuffle e Reduce.

## BIBLIOGRAFIA

- [1] Babak A Farshchian, Thomas Vilarinho, "Which Mobile Health Toolkit Should a Service Provider Choose? A Comparative Evaluation of Apple HealthKit, Google Fit, and Samsung Digital Health Platform", April 2017, LNCS, DOI: 10.1007/978-3-319-56997-0\_12
- [2] INTER-IoT, online, <https://inter-iot.eu/>
- [3] Google Fit, online, <https://developers.google.com/fit/overview>
- [4] Open m-health, online, <https://www.openmhealth.org/>
- [5] Community Health Kit, online, <https://communityhealthtoolkit.org/>
- [6] H. Cai, L. D. Xu, B. Xu, C. Xie, S. Qin and L. Jiang, "IoT-Based Configurable Information Service Platform for Product Lifecycle Management" in IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, vol. 10, no. 2, pp. 1558-1567, May 2014.
- [7] M. T. Lazarescu, "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications" in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 3, no. 1, pp. 45-54, March 2013.
- [8] S. K. Sowe, T. Kimata, M. Dong and K. Zettsu, "Managing Heterogeneous Sensor Data on a Big Data Platform: IoT Services for Data-Intensive Science" 2014 IEEE 38th International Computer Software and Applications Conference Workshops, Vasteras, 2014, pp. 295-300.
- [9] G. Aloï, G. Fortino, R. Gravina, P. Pace and G. Caliciuri, "Edge Computing-Enabled Body Area Networks," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, 2018, pp. 349-353.
- [10] P. Pace, G. Aloï, R. Gravina, G. Caliciuri, G. Fortino and A. Liotta, "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0" in IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 481-489, Jan. 2019.
- [11] Kaur and A. Jasuja, "Health monitoring based on IoT using Raspberry Pi" 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 1335-1340.
- [12] Kim Weins. Cloud Computing Trends: 2017 State of the Cloud Survey. url: [www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2017-state-cloud-survey](http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2017-state-cloud-survey).
- [13] Apache Software Foundation. Apache Hadoop. 2018. url: <http://hadoop.apache.org>.
- [14] Apache Software Foundation. Apache Hadoop powered by. 2018. url: <https://wiki.apache.org/hadoop/PoweredBy>.
- [15] Apache Software Foundation. Apache license-2.0. 2004. url: <http://www.apache.org/licenses/LICENSE-2.0>.
- [16] Sanjay Ghemawat, Howard Gobioff e Shun-Tak Leung. The Google File System. 2003.
- [17] Tom White. Hadoop: The Definitive Guide, 4th Edition. O'Reilly Media, 2004.
- [18] Jeffrey Dean e Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. 2004.
- [19] Apache Software Foundation. Apache Kafka. 2018. url: <https://kafka.apache.org/intro.html>.
- [20] Apache Software Foundation. Apache Spark. 2018. url: <https://spark.apache.org/>.