

UNIVERSITÀ DEGLI STUDI DI ROMA  
TOR VERGATA



FACOLTÀ DI INGEGNERIA  
**Corso di Laurea Magistrale in Ingegneria Informatica**

TESI DI LAUREA IN  
**Ingegneria Informatica**

**4SweepTron: B5G Portable Spectrum  
Monitoring based on Micro-Service  
Architecture**

Relatore:

Prof. **A. Pellegrini**

Correlatore:

Prof. **L. Chiaraviglio**

Laureando:

**Mattia Antonangeli**

Matr. **0311183**

ANNO ACCADEMICO 2022/2023

# Abstract

Questo documento di tesi descrive il sistema IoT SweepTron, i cui nodi di acquisizione sono governati da un'applicazione a microservizi per l'acquisizione, il processamento e l'invio a un db remoto di dati, relativi alla densità di potenza dei campi elettromagnetici (EMF), acquisiti analizzando diverse bande di frequenza pre-5G e 5G configurabili. Inoltre, è possibile effettuare particolari acquisizioni, chiamate *Catture IQ*, che danno informazioni sulla forma d'onda acquisita in termini di fase e quadratura. Dopo aver descritto diverse reti di acquisizione di dati presenti in letteratura, oltre a descrivere in generale l'architettura a microservizi, si descrive il sistema SweepTron, in particolare i nodi SweepTron a livello hardware e software e l'implementazione del server remoto di storage, descrivendo le componenti HW come PC Embedded, analizzatore e antenna, e mostrando i microservizi e come interagiscono tra loro, oltre a descrivere funzionalità aggiuntive necessarie per il corretto funzionamento del sistema. Infine, si mostreranno i risultati ottenuti in SweepTron TITAN, un nodo fisso di misurazione installato al Comune di Bologna. La tesi si conclude descrivendo possibili prospettive future per SweepTron.

# Indice

<b>Abstract</b>	<b>i</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Contesto e Stato dell'Arte</b>	<b>4</b>
2.1 Sistema RATEL . . . . .	4
2.2 Rete di monitoraggio di EMF nazionale . . . . .	6
2.3 Rete di sensori RF Low-Cost . . . . .	7
2.4 Rete di monitoraggio 5G a lungo termine . . . . .	9
2.5 Architettura a microservizi . . . . .	10
2.5.1 Architettura generale basata su microservizi per acqui- sizione di segnali in ambito medico . . . . .	14
2.5.2 Applicazione Web basata su microservizi per dati geo- spaziali . . . . .	19
2.6 IQ Data e catture IQ . . . . .	21
<b>3 SweepTron</b>	<b>27</b>
3.1 Descrizione del sistema . . . . .	27
3.2 Descrizione del software . . . . .	32
3.2.1 Orchestrazione dei servizi . . . . .	33
3.2.2 Sensing microservice . . . . .	36
3.2.3 Processing microservice . . . . .	40
3.2.4 Transfer microservice . . . . .	43
3.2.5 Watchdog microservice . . . . .	45
3.3 Funzionalità aggiuntive . . . . .	47

3.3.1	File di log e funzioni di allarme . . . . .	48
3.3.2	Gestione fase di trasferimento e controllo delle frequenze	48
<b>4</b>	<b>Caso di studio: SweepTron TITAN monitoring node</b>	<b>51</b>
4.1	Risultati . . . . .	55
<b>5</b>	<b>Conclusioni e studi futuri</b>	<b>62</b>
	<b>Ringraziamenti</b>	<b>67</b>

# Elenco delle figure

2.1	Da avs.amazon.com, confronto tra architettura monolitica e architettura a microservizi . . . . .	11
2.2	Descrizione dei diversi stage nell'architettura per il processamento di segnali fisiologici . . . . .	15
2.3	Diagramma dell'architettura a microservizi in ambito medico . .	16
2.4	Architettura lato back-end della PWA . . . . .	20
2.5	Esempio di cattura IQ . . . . .	23
2.6	Funzionamento del buffer circolare per lo streaming di dati IQ .	24
2.7	Streaming in remoto di una cattura IQ . . . . .	25
3.1	SweepTron E2E Breakdown . . . . .	30
3.2	In (a) i componenti del nodo ultraportable: da sinistra, Analizzatore ultraportable Anritsu MS2760A, PC Embedded BPC-3022, Antenna Anritsu 2000-1751-r; in (b) i componenti del nodo fisso: da sinistra, Analizzatore rack Anritsu MS27201A, PC Embedded AVS520, Antenna Steatite Omni QMS-00017 . . . . .	30
3.3	Struttura nodo SweepTron . . . . .	33
3.4	Scenario tipico di comunicazione in fase di trasmissione . . . . .	34
3.5	Flowchart microservizio di sensing . . . . .	37
3.6	Flowchart microservizio di processing . . . . .	42
3.7	Flowchart microservizio di transfer . . . . .	44
3.8	Flowchart microservizio di watchdog . . . . .	46
3.9	Esempio di file di log. In questa figura trattasi di log del microservizio di Sensing. . . . .	49

3.10	Esempio di file di log relativo agli errori. Utilizzato per funzioni di allarme. . . . .	49
4.1	Band Plan italiano per bande 4G e 5G . . . . .	52
4.2	Foto del nodo TITAN installato al Comune di Bologna. La centralina Anritsu MS27201A e il PC Embedded AVS 520 che esegue i servizi sono installati all'interno del rack in figura, e alimentati dall'edificio. Il PC Embedded è collegato a un router LTE con antenna posta all'interno del rack, mentre l'antenna STEATITE è posta al di fuori dell'edificio ed è collegata all'analizzatore rack tramite cavo coassiale, che passa attraverso un foro al muro che divide rack e antenna. . . . .	53
4.3	Software Anritsu MS27201A per Windows, in cui viene mostrata la configurazione della misurazione che si sta effettuando dal punto di vista di banda, unità di misura, finestra di visualizzazione, markers, e così via. Nel caso dell'immagine, si tratta del nodo TITAN, l'unità di misura dei dati acquisiti è $dBm/m^2$ e la banda analizzata è la 758-768 MHz. . . . .	54
4.4	Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi nelle diverse bande in $dBm/m^2$ . . . . .	58
4.5	Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi, con comparazione per le bande subGHz e 3.5GHz, in $dBm/m^2$ . . . . .	59
4.6	Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi per i diversi operatori in $dBm/m^2$ . . . . .	60

# 1. Introduzione

L'Internet of Things (IoT), o Internet delle Cose, che consiste in una rete globale di sistemi interconnessi, è sempre più diffuso in ambiti aziendali e industriali, e la sua diffusione va di pari passo con la crescita delle tecnologie digitali che sostengono le attività sociali ed umane. Infatti, ci sono sempre più dispositivi connessi ad Internet automatizzati per l'acquisizione di dati di vario tipo che vengono utilizzate per migliorare la vita delle persone grazie a innovazioni rivoluzionarie, come l'agricoltura smart, la guida autonoma dei veicoli, l'industria 4.0, ma anche tecnologie per evitare sprechi di acqua, diminuire il riscaldamento globale, e così via.

Quando si parla di IoT, sono importanti le fasi di acquisizione, elaborazione e trasferimento di dati via Internet. Con il passare del tempo, l'aumento delle capacità computazionali dei sistemi e la diminuzione dei prezzi delle memorie secondarie, oltre all'aumento delle larghezze di banda di rete grazie alle nuove tecnologie come il 5G, ha fatto sì che il mondo dell'IoT si evolvesse in maniera considerevole.

È quindi importante la gestione della connettività wireless, sia lato SW che lato HW. Al fine di migliorare le prestazioni di rete, sono state costruite diverse infrastrutture di rete wireless, come GSM, UMTS, o sistemi Wi-Fi che, ovviamente, generano campi elettromagnetici (EMF) per la trasmissione di dati. Allo stesso modo, è altamente di interesse tenere conto dell'esposizione umana agli EMF, che da sempre incontra lo scetticismo della popolazione per quanto riguarda eventuali effetti sulla salute che EMF ad alta frequenza possono causare, nonostante siano stati effettuati diversi studi scientifici che non mostrano evidenze di questi effetti. L'obiettivo della tesi proposta è quello di definire e

realizzare un sistema di monitoraggio degli EMF al fine di acquisire informazioni utili sull'utilizzazione di canali wireless a diverse bande di frequenza, in particolare per l'acquisizione di dati a lungo termine, al fine di valutare l'evoluzione delle diverse tecnologie di trasmissione e ricezione utilizzate oggi (e.g. 4G, 5G).

Citando il Malta Communications Authority (MCA) [12], "*The purpose of spectrum monitoring is to support the spectrum management process in general, including frequency assignment and planning functions*". Le tecnologie di reti cellulari si sono evolute in maniera importante negli ultimi anni, fino a diventare praticamente ubiqui nella vita delle persone. Attualmente, i servizi pre-5G e 5G sono ampiamente diffusi in diversi paesi, incluso il territorio italiano. Diverse reti di sensori sono state progettate e dispiegate nel mondo (e.g. in Belgio, Francia, Grecia, Italia, Malta, Portogallo, Africa), al fine di effettuare un monitoraggio spaziotemporale di campi elettromagnetici in radiofrequenza (RF-EMF), e quindi acquisire dati su diverse bande pre5G/5G tramite analizzatori di spettro.

Il monitoraggio di spettro è importante per effettuare un controllo delle emissioni di EMF per verificare la conformità alle licenze, recuperare informazioni sull'utilizzo effettivo dello spettro, assistenza nell'indagine delle interferenze radio su scala locale, regionale e globale, individuazione e identificazione di trasmettitori illegali, identificazione e misurazione di segnali interferenti.

I dati misurati, dunque, possono essere utili a diversi scopi: misurazioni a lungo periodo possono essere utilizzati per prevedere l'andamento futuro (crescita/decrecita dell'utilizzazione) delle tecnologie pre5G/5G; misurazioni sulla potenza dei EMF sono importanti per capire quanto le persone siano esposte a campi elettromagnetici che potrebbero essere dannosi per la salute, e verificare che le potenze dei EMF siano al di sotto dei limiti imposti dallo stato.

In questo lavoro viene presentato SweepTron, un nuovo sistema di monitoraggio basato su un'architettura a microservizi. L'obiettivo di questo lavoro è quello di definire una nuova architettura, scalabile e facilmente integrabile,

che permetta di:

- Definire una rete di monitoraggio in larga scala;
- Effettuare misurazioni a diverse bande pre5G/5G facilmente configurabili;
- Raggruppare i risultati in un database spaziotemporale per analisi successive;

Oltre alle misurazioni di potenza di EMF, SweepTron offre la possibilità di effettuare particolari misurazioni di onda chiamate "catture IQ". Questa tipologia di cattura permette di acquisire, con un livello di dettaglio molto alto, informazioni relative alla forma d'onda misurata in tempo reale, e permette di salvarne le informazioni in locale.

I risultati salvati da SweepTron possono essere utilizzati per vedere in prima battuta l'utilizzazione e l'esposizione delle diverse bande analizzate, oltre all'analisi della qualità del segnale, ma acquisendo dati per un periodo sufficientemente lungo si può evidenziare la presenza di trend secolari sull'insieme di dati ottenuti, e quindi avere informazioni importanti sulla crescita o decrescita dell'utilizzazione di diverse bande, oppure delle tecnologie di rete quali 4G e 5G. Queste informazioni possono essere utili per comprendere e prevedere l'evoluzione di reti future, come ad esempio le reti 6G.

## 2. Contesto e Stato dell'Arte

In questo capitolo si descriveranno le diverse idee su reti di monitoraggio di EMF introdotte da altre ricerche in ambito accademico e si analizzeranno gli aspetti di forza e le migliorie che il progetto SweepTron apporta alle reti citate. Inoltre, si introdurranno concetti di architetture a microservizi, descrivendo sistemi di raccolta dati basati su di essi, in modo da comprendere la successiva applicazione di questa architettura alla rete di acquisizione SweepTron.

### 2.1 Sistema RATEL

Il sistema RATEL [7] è una rete di monitoraggio di EMF creata nella Facoltà di Scienze Tecniche, Università di Novi Sad, Novi Sad, Serbia. L'obiettivo di questa rete consiste nell'acquisizione di dati relativi alla potenza degli EMF per diverse bande al fine di controllare l'esposizione umana a EMF generati dalle nuove tecnologie (come ad esempio 5G), sia a lungo termine, sia a breve termine. In particolare, con i dati acquisiti, si verifica che l'esposizione umana a questi EMF non superi le soglie definite dall'International Commission for Non-Ionizing Radiation Protection (ICNIRP) [10], che ha pubblicato le "Guidelines for limiting exposure to electromagnetic fields (100 kHz to 300 GHz)", ICNIRP 2020 [9], in cui vengono riassunte tutte le guide linea e le informazioni scientificamente provate sugli effetti degli EMF sulla salute.

La rete RATEL è composta da diverse stazioni di monitoraggio di EMF autonome e distribuite geograficamente nel territorio serbo per la misurazione. In particolare, sono stati acquisiti dati per 55 diverse città serbe. I dati acquisiti per ciascun nodo vengono salvati localmente nel nodo e inviati periodicamente,

via rete GSM, al RATEL Database [6], un database centralizzato che definisce diverse tabelle per mantenere informazioni su file, misurazioni, perturbazioni misurate e allarmi causati da eventuali problemi di misurazione. Questi dati vengono poi resi pubblici agli utenti tramite un portale via Internet.

Le misurazioni vengono effettuate utilizzando tre diversi tipi di antenne, su specifiche bande di frequenza e misurazione isotropica. Inoltre, sono stati considerati due casi di studio per la misurazione:

- Wideband EMF Monitoring: viene monitorata l'intera banda da 100kHz a 7/8GHz (in base all'antenna utilizzata per la misurazione) con l'obiettivo di calcolare la potenza di EMF totale nell'intervallo considerato;
- Service-Based EMF Monitoring: viene monitorata la banda da 100kHz a 6GHz, ma con un'antenna capace di dividere le misurazioni in 20 sottointervalli di frequenze programmabili. In questo modo, è possibile calcolare il contributo di potenza di EMF per ogni sottointervallo.

Le bande monitorate sono state definite in base al Regulatory Agency for Electronic Communications and Postal Services [16], in particolare considerando l'allocazione dello spettro nella Repubblica di Serbia.

I risultati ottenuti dalla misurazione e le relative analisi sono consultabili su un portale del sito di RATEL [15].

In SweepTron, il monitoraggio è analogo al secondo caso di studio visto sopra, ma la divisione del monitoraggio in diverse bande è totalmente gestito lato software. Come si vedrà nei prossimi capitoli, le bande vengono monitorate ciclicamente con un approccio Round Robin. Inoltre, nel sistema RATEL, i nodi dei sensori effettuano solamente operazioni di salvataggio dei dati e invio periodico al server, a differenza dei nodi SweepTron che, oltre ad offrire meccanismi di compressione dei dati, permettono di effettuare particolari catture (*IQ capture*) molto dettagliate sulla forma d'onda misurata, al fine di poter effettuare analisi successive molto precise.

## 2.2 Rete di monitoraggio di EMF nazionale

Questo paragrafo descrive la rete di monitoraggio EMF pubblica italiana [19], costruita dal Ministero delle Comunicazioni Italiano in collaborazione con la Fondazione Ugo Bordoni (FUB) e con le Agenzie di Protezione dell'Ambiente locali di tutte le regioni italiane. Il progetto è nato nel 2002 ed è ufficialmente terminato nel 2006, lasciando il posto alla campagna BluShuttle [19], di diversa natura rispetto alle reti di monitoraggio che stiamo considerando, in quanto utilizzati principalmente per brevi campagne di comunicazione (non più di mezz'ora) utilizzando piccole automobili che montano misuratori di EMF Wideband autonomi.

La rete è stata distribuita nel territorio al fine di "*contrastare la diffusa paura, tra l'opinione pubblica nazionale, di pericoli per la salute sconosciuti dovuti all'esposizione ai campi elettromagnetici*" [19]. In particolare, l'istituzione di questa rete ha molteplici obiettivi:

- Informare il pubblico sulle attuali conoscenze scientifiche, che non mostrano evidenze su effetti negativi sulla salute dovuti all'esposizione ai EMF;
- Informare il pubblico sulla normativa italiana sulle emissioni di EMF, tra le più restrittive al mondo;
- Dimostrare che i limiti imposti dalla normativa italiana sulle emissioni di EMF sono ampiamente rispettati;
- Attivare procedure al fine di ridurre i livelli di esposizione quando i suddetti limiti vengono superati.

Da un punto di vista di progettazione, la rete è definita come un'architettura decentralizzata: diverse stazioni di misurazione remote (EMF Stations) acquisiscono dati che vengono inviati, via rete GSM, a dei centri di controllo locali dispiegati su base regionale o sub-regionale. Da qui, i dati viaggiano verso

il Centro di Archiviazione Nazionale che si trova presso i locali del Ministero delle Comunicazioni.

Per il rilevamento degli EMF, sono state utilizzate sia sonde wide-band, sia sonde multi-band, in modo da effettuare misurazioni di tutte le fonti di radiofrequenza che operano tra 100kHz e 3GHz. In particolare, le sonde wide-band sono utilizzate per verificare il rispetto dei limiti di esposizione su tutta la banda considerata. Invece, le sonde multi-band sono utili per misurazioni divise in base alla classe della sorgente (trasmissioni radio/TV, mobile BS) e i dati ottenuti vengono utilizzati dalle Environmental Protection Agencies per l'interpretazione e la validazione dei dati misurati.

Focalizzandosi sulle stazioni di monitoraggio, esse sono progettate in modo da essere leggere, compatte, facilmente trasportabili ed autonome. Infatti, esse dispongono di pannelli fotovoltaici e sono equipaggiate con modem GSM in modo da poter comunicare autonomamente i dati misurati ai centri di controllo locali, segnalare eventuali problemi con segnale di allarme, ma anche ricevere configurazioni da remoto tramite downlink.

I centri di controllo remoti consistono di un PC dotato di un software ad-hoc, che comunica con le stazioni di monitoraggio, riceve informazioni da esse che, previa validazione di personale qualificato, vengono inviate via email allo staff del FUB, che effettuano poi upload dei dati al Centro di Archiviazione Nazionale. I dati così ottenuti sono poi stati resi pubblici su un sito web relativo al progetto, ora chiuso.

### 2.3 Rete di sensori RF Low-Cost

Questo paragrafo descrive il design e lo sviluppo di una rete di nodi di sensori RF low-cost distribuita in Antwerp, Belgio [1]. Questa rete è stata ideata per mostrare le potenzialità delle infrastrutture IoT. Infatti, con l'avvento dell'IoT, è stato possibile effettuare il deployment di dispositivi e sensori che possono essere anche meno complessi e costosi, al fine di misurare diversi parametri ambientali autonomamente e per lunghi periodi.

La rete di sensori RF è stata progettata con componenti off-the-shelf senza studio di design particolare per ogni componente, in modo da minimizzare i costi; è in grado di effettuare misurazioni su quattro bande di frequenza e ad alta risoluzione temporale, tramite quattro antenne che operano sulle frequenze considerate (in particolare, reti cellulari GSM, UMTS, LTE, e reti Wi-Fi). In particolare, si osservano due diverse reti di monitoraggio:

- Rete *fixed*: 10 dispositivi sono diffusi in una piccola area geografica nel centro della città di Antwerp; la rete effettua misurazioni ad alta risoluzione temporale. I dispositivi facenti parte di questa rete sono collegati fisicamente al gateway ed alimentati dagli edifici specifici su cui sono montati. Questi nodi effettuano misurazioni in un'area fissa e predeterminata al momento del montaggio, in modo che il sampling time sia di un secondo e salvano le misurazioni su un database centrale.
- Rete *mobile*: 4 dispositivi sono installati su veicoli del servizio postale locale; tale rete tende a coprire un'area geografica molto più ampia, seppur con misurazioni meno fitte. I nodi mobili sono collegati al gateway tramite long-range wide area network (LoRaWAN) e sono alimentati dal veicolo su cui sono montati. Questi nodi effettuano misurazioni in modo che il sampling time sia di 30 secondi (per evitare un carico troppo alto sulla rete LoRaWAN, che ha transmission rate bassi).

È interessante vedere come una rete low-cost, se definita nei minimi dettagli, sia in grado di effettuare misurazioni che, seppur grossolane, ci danno delle informazioni generali importanti per impostare studi futuri. SweepTron, a differenza della rete sopra introdotta, utilizza analizzatori di spettro più potenti e con svariate possibilità di misurazione, oltre alla già citata cattura IQ che permette di effettuare analisi estremamente dettagliate sulle forme d'onda catturate.

## 2.4 Rete di monitoraggio 5G a lungo termine

In [5] viene definito un sistema di monitoraggio delle reti 5G il cui obiettivo non è unicamente quello di acquisire dati sull'esposizione nel territorio, ma anche quello di capire e caratterizzare l'evoluzione dell'esposizione delle reti 5G per tempi molto lunghi (mesi/anni). Il monitoraggio a lungo termine, oltre all'individuazione di trend secolari che permettono, ad esempio, di comprendere l'impatto dell'utilizzazione delle reti 5G da parte degli utenti sull'esposizione, ci permettono di comprendere l'evoluzione della rete 5G stessa (e.g. cambio di frequenze da parte di operatori, uso di nuove bande per 5G).

Il sistema di monitoraggio realizzato per raggiungere lo scopo sopra citato permette di acquisire informazioni sull'esposizione 5G e di compararle con l'esposizione pre-5G. In particolare, un nodo effettua misurazioni utilizzando il dispositivo Anritsu MS27102A, un analizzatore di spettro in grado di monitorare lo spettro di frequenze nell'intervallo 9kHz–6GHz, a cui è collegata via cavo coassiale un'antenna passiva omnidirezionale Keysight N6850A. L'analizzatore è collegato via LAN e controllato da un PC General Purpose, in particolare dal software Search24 [5].

Search24 è un software proprietario di ARPA Lazio scritto in Matlab, che tramite *Standard Commands for Programmable Instruments* (SCPI), permette di acquisire diverse informazioni in base alle impostazioni (e.g. frequenza centrale, span, numero di samples, e così via).

Il nodo di misurazione così costruito è stato installato al quarto piano di un edificio universitario di Roma Tor Vergata, ed effettua misurazioni sulle reti pre-5G/5G appartenenti a Wind/3. La scelta è motivata dal fatto che il nodo si trova a circa 580m dalla Base Station servente, oltre al fatto che essendo una zona universitaria il traffico è potenzialmente non trascurabile. Inoltre, dal punto di vista di Line Of Sight (LOS), essendo il nodo situato al quarto piano, ha buone condizioni di copertura e di propagazione, e quindi potenzialmente buoni livelli di segnale rispetto a tutte le tipologie di reti.

Dai dati acquisiti (Novembre 2021–Maggio 2022) si evince che i segnali

pre-5G dominano dal punto di vista di esposizione, ma nel tempo si vede che l'esposizione dei segnali 5G tende a crescere a discapito dei segnali pre-5G. In particolare, si nota questo comportamento studiando i picchi di segnale 5G ottenuti in termini di esposizione.

Il progetto SweepTron è nato sulla base di quest'ultimo articolo. L'obiettivo è stato infatti quello di realizzare un software basato su una architettura a microservizi che potesse comunicare con un analizzatore di spettro Anritsu, acquisire dati sulla potenza di EMF, e trovare dei risultati simili a quelli sopra, aggiungendo diverse caratteristiche utili per la gestione in remoto dei nodi di misurazione, come ad esempio la gestione degli errori e degli allarmi.

## 2.5 Architettura a microservizi

Essendo SweepTron realizzato utilizzando uno stile architetturale a microservizi, è importante aprire un discorso su questa tipologia di architettura. L'architettura a microservizi viene descritta in letteratura come "*a pattern for implementing business logic within an organization using small, single-purpose services*" [11]. In particolare, questo stile architetturale è in netta contrapposizione con l'architettura monolitica tradizionale, caratterizzata da un'unica applicazione per eseguire più funzioni aziendali interdipendenti (come un'interfaccia utente lato client, un database e una applicazione lato server).

La Figura 2.1 mostra un esempio grafico di implementazione di tre diversi servizi. Nell'approccio monolitico, i componenti logici di un sistema sono tutti posti all'interno della stessa applicazione, ed essi sono fortemente accoppiati e interdipendenti, mentre l'approccio a microservizi definisce tre applicazioni diverse, una per servizio offerto, e sono quindi più lascamente accoppiati e potenzialmente indipendenti.

Per capire a fondo i motivi per cui è stata introdotta l'architettura a microservizi, riassumiamo le caratteristiche delle applicazioni monolitiche [2][3]. L'architettura a microservizi porta diversi vantaggi, ma non per questo bisogna ignorare l'approccio "tradizionale". Infatti, anch'esso ha dei punti di forza:

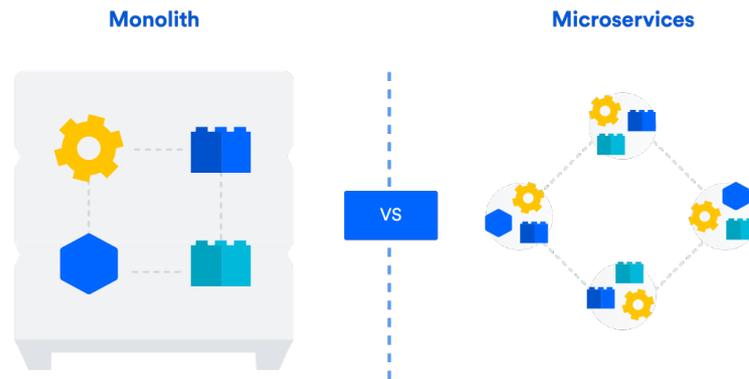


Figura 2.1: Da [aws.amazon.com](https://aws.amazon.com), confronto tra architettura monolitica e architettura a microservizi

- Essendo le componenti logiche implementate in una applicazione singola, il deployment di un'applicazione monolitica può risultare più semplice, dato che bisogna configurare un unico ambiente di esecuzione, mentre nel caso di applicazioni a microservizi vanno impostati più ambienti di esecuzione, di solito gestiti tramite container;
- Da un punto di vista di sviluppo, questa architettura può essere più efficiente in caso di applicazioni relativamente piccole e dallo sviluppo rapido, poiché non ha bisogno di una fase di pianificazione intensa come nel caso delle applicazioni a microservizi, in cui va pianificata nei minimi dettagli l'orchestrazione dei servizi;
- Le fasi di testing e debugging sono più semplici, essendo l'applicazione composta da una singola unità centralizzata;

Tuttavia, gli svantaggi sono oggi troppo evidenti, soprattutto considerando il mondo Cloud in cui siamo immersi:

- Lo sviluppo di grandi applicazioni può essere lento e complesso.
- Non è possibile scalare componenti individuali, ma se necessario si deve scalare l'intera applicazione, il che può causare uno spreco di risorse computazionali e di memoria.

- Poco flessibile rispetto ai linguaggi e alle tecnologie utilizzate. Un cambio in tal senso porta a costi e tempi di sviluppo molto elevati.
- Cambiamenti nell'applicazione richiedono un redeployment dell'intera applicazione.

Si comprende bene come gli svantaggi di questa architettura vadano in conflitto con il mondo del cloud computing, soprattutto dal punto di vista della scalabilità e della flessibilità delle applicazioni. A questo proposito, una caratteristica importante dell'architettura a microservizi è data dal fatto che i servizi così costruiti sono tendenzialmente piccoli, debolmente accoppiati e distribuibili in modo indipendente. Infatti, normalmente questi servizi hanno un proprio stack tecnologico (compresa gestione dei dati e database), comunicano tra loro via API REST, eventi e message broker, e offrono benefici importanti anche a livello aziendale e organizzativo:

- Permettono un aggiornamento del codice relativo a uno specifico servizio in maniera semplificata e indipendente (se si vuole aggiungere una funzionalità a un microservizio, solo la sua applicazione va modificata e ricostruita).
- Si possono utilizzare diversi linguaggi e tecnologie per i diversi servizi, scegliendoli in modo opportuno in base alle funzionalità che essi devono offrire.
- Possono essere scalati indipendentemente l'uno dall'altro. Questa caratteristica è molto importante soprattutto in ambito cloud, poiché è possibile scalare soltanto il servizio che ne ha bisogno, evitando quindi di replicare i servizi di bassa utilizzazione.
- Il deployment dei servizi può essere fatto in maniera indipendente.
- I servizi, presi singolarmente, sono altamente mantenibili e testabili. È semplice isolare un singolo servizio ed effettuare operazioni di testing e debugging su di esso.

Ovviamente, l'architettura a microservizi non è perfetta, ma anch'essa ha degli svantaggi di cui tenere conto:

- Un'applicazione per ogni servizio possono essere costruiti anche da diversi team di sviluppo. Questo può portare quindi a uno sviluppo più lento e a performance operazionali minori.
- I costi di infrastruttura aumentano sensibilmente a causa di infrastrutture di hosting, tool di monitoraggio, test suite, e così via.
- Per la comunicazione dei servizi e il coordinamento viene aggiunto dell'overhead dal punto di vista organizzazionale.
- Diversi linguaggi e tecnologie portano a una mancanza di standardizzazione, e diversi servizi portano a una mancanza di ownership chiara dei servizi stessi da un punto di vista di team di sviluppo.

Riassumendo, mentre l'architettura monolitica può essere una buona soluzione per applicazioni relativamente semplici e centralizzate, essa non si presta bene in caso di applicazioni distribuite, soprattutto in ambito cloud. L'architettura a microservizi permette un deployment più elastico e flessibile in questi ambienti, ma porta anche dei vantaggi dal punto di vista di flessibilità nell'utilizzo di linguaggi per lo sviluppo e l'aggiornamento dei servizi per l'aggiunta di nuove funzionalità.

Se analizziamo i nodi SweepTron, essi eseguono un software costruito a microservizi. Si è scelto questo approccio per sfruttare i vantaggi descritti sopra, in particolare i vantaggi relativi alla semplicità di sviluppo e manutenzione dei singoli servizi, oltre alla possibilità di aggiungere in futuro nuove funzionalità. Di fatto, il software presentato vuole essere una buona base su cui costruire in futuro altre feature, come la compatibilità con altri strumenti di misura, la possibilità di effettuare diversi tipi di catture e di processamento di dati anche in edge, e così via. Allo stesso modo, il server remoto che immagazzina i dati è attualmente centralizzato, ma può essere in futuro implementato in maniera distribuita a fine di migliorarne l'affidabilità, la disponibilità, la resistenza ai

guasti e evitando situazioni di collo di bottiglia in caso di elevato traffico in ricezione.

In letteratura sono presenti degli studi riguardanti reti di sensori in cui l'architettura a microservizi è utilizzata per diversi motivi. Essi sono descritti nelle prossime sezioni.

### **2.5.1 Architettura generale basata su microservizi per acquisizione di segnali in ambito medico**

In [17] viene proposta una architettura generale basata su microservizi e event-driven al fine di migliorare l'acquisizione dei dati relativi a segnali fisiologici (e.g. battito cardiaco, pressione del sangue, temperatura corporea, saturazione dell'ossigeno). In particolare, l'obiettivo di questa architettura è quella di assicurare accuratezza e affidabilità dei dati misurati, minimizzando i delay e gli errori di acquisizione dei dati in modo da avere una caratterizzazione della salute del paziente il più accurata possibile.

È evidente che questa applicazione è tipica del mondo IoT: diversi sensori vengono installati sul corpo del paziente (e.g. dispositivi indossabili, ma anche sensori installabili all'interno del corpo umano), e diverse informazioni vengono acquisite da essi. I dati devono poi essere inviati in un centro di controllo, che permette, dopo opportune analisi, di comprendere le condizioni di salute del paziente. L'acquisizione dei dati, se accurata e real-time, permette di identificare potenziali problemi di salute di un paziente prima che essi diventino più seri, in modo da poter intervenire in anticipo con cure specifiche e, quindi, avere cura del paziente in maniera più specifica.

La Figura 2.2 descrive le diverse fasi che intercorrono dall'acquisizione allo storage al report dei dati. In particolare, possiamo vedere:

- Stage 0: Devices e sensori: rappresenta il livello HW, ovvero i sensori addetti all'acquisizione dei segnali fisiologici.
- Stage 1: Acquisition: definisce il layer SW che interagisce con i sensori a livello più basso. Riceve quindi dati grezzi, da raffinare successivamente.

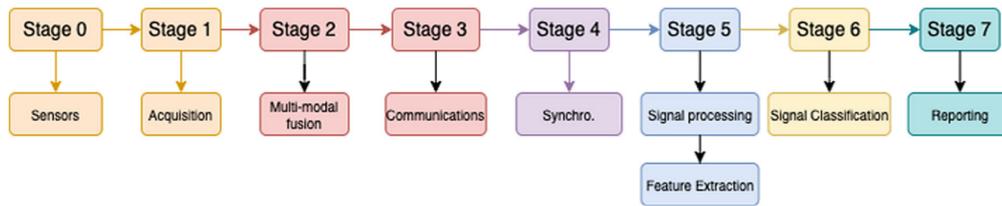


Figura 2.2: Descrizione dei diversi stage nell'architettura per il processamento di segnali fisiologici

- Stage 2: Multi-modal fusion: sincronizzazione dei diversi flussi di dati provenienti dai diversi sensori. In questa architettura la sincronizzazione può essere complicata a causa della latenza di rete e da un punto di vista di comunicazione. Per questo motivo è necessario uno stage apposito per questa gestione.
- Stage 3: Communications: layer di comunicazione tra il layer di sincronizzazione e il resto dell'infrastruttura. Spesso si usano protocolli come Laboratory Stream Layer (LSL) e Message Queuing Telemetry Transport (MQTT) per la fase di comunicazione. In particolare, MQTT è un protocollo leggero di tipo publish-subscribe usato per reti low-bandwidth, high-latency e non affidabili, e facilmente integrabile con diversi dispositivi e piattaforme; per questo motivo si presta bene nel caso di comunicazione tra diversi dispositivi e sensori.
- Stage 4: Synchronization Strategy: altro fattore critico, dato che siamo nel mondo dei microservizi. In generale, vengono definite due sincronizzazioni e ambienti di storage: ambiente locale ed ambiente cloud. Nel primo, avviene l'acquisizione dei dati e l'invio di essi ai protocolli di comunicazione per essere ricevuti dall'ambiente cloud. Nel secondo, i dati vengono ricevuti dal primo e vengono resi disponibili per processamenti successivi. Il tutto è generalmente gestito con un Message Broker (MB) che agisce come intermediario (e.g. MQTT).
- Stage 5: Processing and Feature Extraction: tendenzialmente diviso in una fase di pre-processing in cui vengono filtrati i dati escludendo ad

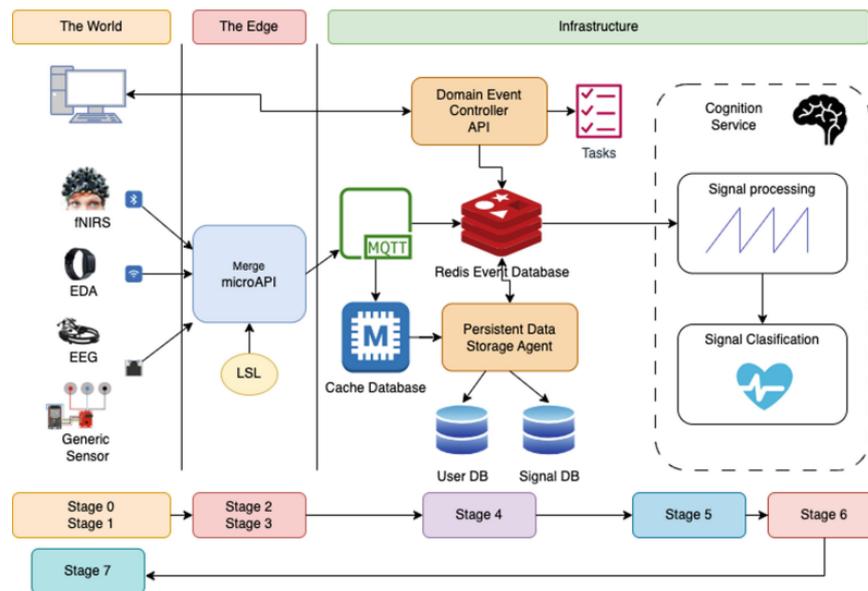


Figura 2.3: Diagramma dell'architettura a microservizi in ambito medico

esempio gli outlier, una fase di processing in cui si applicano dei metodi specifici in base al tipo di segnale, e infine una fase di feature extraction in cui l'estrazione si basa sulla dipendenza delle variabili dal tempo o dalla frequenza, parametri statistici o morfologia del segnale.

- Stage 6: Signal Classification: uso di modelli preaddestrati che possono distinguere diverse situazioni, in base ai segnali (processati) ottenuti in input. Questa fase è importante soprattutto in ambito medico, dal momento che modelli di Machine Learning sono sempre più utilizzati per effettuare diagnosi anche molto importanti, come la presenza di tumori o condizioni patologiche gravi.
- Stage 7: Reporting: creazione e uso di una User Interface (UI) per mostrare all'utente i dati processati e i risultati, in modo da effettuare studi futuri sulle condizioni dei pazienti.

Dopo aver elencato i diversi stage da svolgere per l'acquisizione e l'analisi dei dati, l'articolo propone anche (Figura 2.3) una descrizione dell'architettura a microservizi che possa svolgere le fasi di cui sopra. In particolare, vediamo che le fasi relative all'acquisizione dei dati sono associate a una "zona di ac-

quisizione" chiamata The World. I sensori sono connessi con diverse tecnologie (e.g. Bluetooth, WiFi) alla zona chiamata The Edge, addetta alle fasi 2 e 3. La fase di Multi-modal fusion è gestita da una micro-API (application programming interface) che, usando LSL, sincronizza i diversi tipi di data streams (e.g. ECG, dati di motion, EEG) in real time con alta precisione. Si passa alle fasi successive via comunicazione con MQTT, storage di dati, processamento dei dati e classificazione, il tutto nella zona di infrastruttura. Il reporting restituisce i dati in output nella World zone.

L'architettura proposta in questo articolo è molto interessante, ed è studiata per gestire al meglio diversi flussi di dati derivanti da diversi sensori di tipo fisiologico, oltre ad introdurre una fase di classificazione di segnali, importantissimo in ambito medico. Se volessimo confrontare questa architettura con quella presentata da SweepTron, ci sono dei punti di similarità e delle differenze.

Se guardiamo unicamente il nodo SweepTron, esso è stato studiato per effettuare misurazioni da una singola fonte per volta, poiché è un limite dell'analizzatore che si sta utilizzando. Il merge dei flussi, dunque, viene effettuato al livello del server remoto, che nel momento in cui salva i dati all'interno del database spaziotemporale li etichetta in base al nodo da cui ha ricevuto i dati. Inoltre, non viene effettuata classificazione dei segnali con modelli di Machine Learning, ma vengono utilizzati meccanismi di conversione al fine di ottenere diverse informazioni a partire dalle catture effettuate dall'analizzatore. A questo proposito, la modularità del software di SweepTron è molto utile: le funzioni di conversione possono essere aggiunte in maniera semplice, sia a livello di nodo che a livello di Cloud nel server remoto, il che ci dà molte possibilità di personalizzazione e scelta anche in base a eventuali profiling del sistema da un punto di vista di rete e capacità computazionali.

Similmente a quanto analizzato sopra, in [18] viene descritto un sistema di acquisizione e sincronizzazione di dati fisiologici, in cui i dispositivi di acquisizione sono di due tipi:

- Empatica E4 Wristband: dispositivo da polso in grado di acquisire diversi

biosegnali, come pressione sanguigna, battito cardiaco, temperatura, il tutto in real time via Bluetooth. Il dispositivo deve essere saldamente collegato al polso per evitare problemi di acquisizione di biosegnali.

- **Wireless Electrocardiogram AD8232 Device:** un sensore che viene collegato tramite elettrodi al petto, all'altezza del cuore, al fine di misurare l'attività elettrica del cuore per un certo periodo di tempo. Il sensore ha un sampling rate di 360Hz e permette monitoraggio continuo e prolungato grazie a un basso consumo di potenza. Per garantire connettività WiFi e Bluetooth, è collegato a un microcontrollore ESP32, oltre ad avere diversi pin analogici digitali per aggiungere diversi sensori.

Come già detto, l'architettura definita in questo articolo è una specializzazione di quella definita precedentemente. In particolare, sono specificate le diverse tecnologie utilizzate per i diversi servizi. Possiamo vedere infatti come venga utilizzato Kubernetes per l'orchestrazione dei diversi servizi di API Server, Web Server, oltre a servizi di collection, processamento, ordinamento e storage delle metriche acquisite.

Allo stesso modo, viene descritta nei minimi dettagli la fase di sincronizzazione dei diversi flussi di dati acquisiti, in questo caso, dai due dispositivi Empatica E4 e AD8232. Esso viene fatto dividendo l'ambiente locale in due parti:

- **Sensing zone:** i dispositivi acquisiscono dati e si collegano all'edge tramite un "Sensor Hub", che può gestire diverse tecnologie di connettività (Bluetooth o WiFi)
- **The edge:** un certo numero di agents si occupano di preparare i dati assegnando un timestamp e segnalandoli al resto dell'infrastruttura via protocollo MQTT.

Successivamente, i dati vengono ri-taggiati con un timestamp ottenuto da NTP per una verifica di latenza, vengono salvati in un Cache DB per motivi di lentezza di scrittura, e un evento, che specifica lo stato del segnale ricevuto

da un punto di vista di processamento e storage, viene salvato nell'Event DB. Infine, i dati vengono processati e classificati da una serie di microservizi in base al contesto.

Da un punto di vista di risultati, è interessante vedere come questo sistema presenta una latenza minore rispetto a una sincronizzazione con connessione diretta al cloud. Questo è dovuto ovviamente alla latenza di rete, che viene evitata effettuando sincronizzazione in locale. In particolare, per i sensori testati, si è ottenuta una latenza tra i 50 e i 100 ms per il processamento dei dati, il che indica che sia possibile effettuare implementazioni real-time di questo tipo per qualsiasi sistema e sensore.

### 2.5.2 Applicazione Web basata su microservizi per dati geospaziali

In [13] viene descritto un sistema che permette di visualizzare dati relativi a misurazioni geospaziali, tramite una Progressive Web App costruita usando un approccio a microservizi, al fine di integrare diverse tecnologie e metodi di orchestrazione per combinare diverse informazioni geospaziali e IoT. La web app è costruita, da un punto di vista di UI, per permettere di gestire e integrare i dispositivi IoT in una UI component-based, oltre a mostrare modelli di dati graph-based con un algoritmo di selezione in grado di recuperare informazioni da componenti specifiche in base al contesto utente.

Nella Figura 2.4 è rappresentata la struttura e i servizi della parte back-end dell'applicazione. In particolare, vediamo come i microservizi siano suddivisi in tre gruppi:

- Edge Services: layer in cui l'obiettivo principale è l'orchestrazione delle comunicazioni tra i microservizi. Vengono utilizzate delle componenti Netflix OSS che permettono di implementare meccanismi e pattern specifici per i microservizi, come Circuit Breaker, API Gateway e Discovery lato client e server. In questo layer di servizio, le richieste vengono orchestrate a livello di microservizio, e le richieste esterne vengono re-

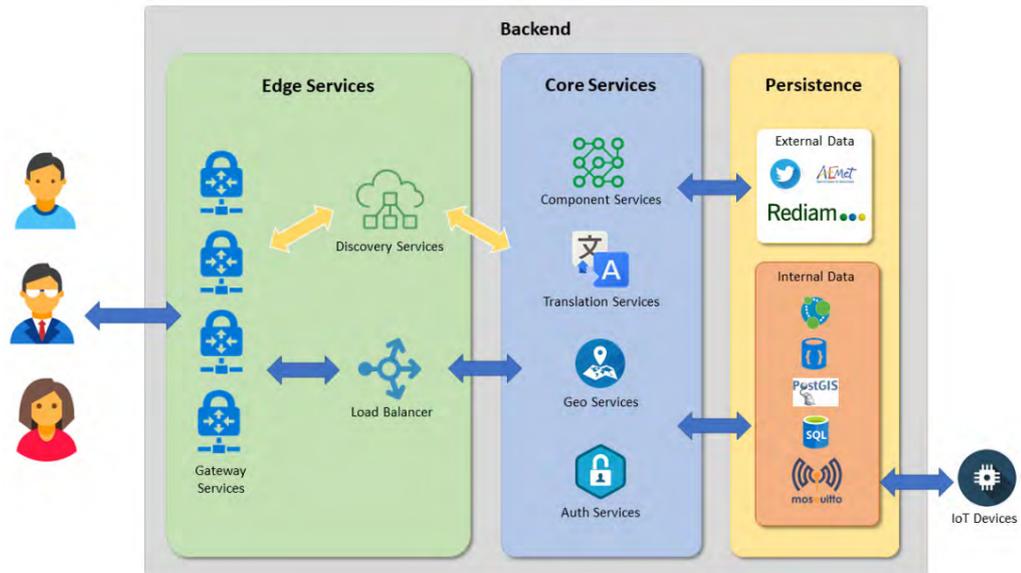


Figura 2.4: Architettura lato back-end della PWA

direzionate opportunamente al core layer. Inoltre, questi servizi sono replicabili a livello di container in base alle esigenze di carico: dato un certo numero di container relativi a uno specifico servizio, se il carico di richieste che essi devono gestire è troppo alto, si effettua lo spawn di nuove repliche in modo da poter fronteggiare carichi più elevati.

- **Core Services:** implementano la business logic dell'applicazione, divisa in diversi servizi: l'Auth Service gestisce autenticazioni e autorizzazioni degli utenti, il Translation Service offre meccanismi di traduzione in base al contesto utente, sia di lingua che di unità di misura dei dati. Il Geo Service è un servizio di middleware che gestisce richieste geospaziali. Il Component Service gestisce le richieste dell'utente, e fa in modo che l'utente riceva i dati specificatamente richiesti da una certa componente o insieme di componenti associati a una certa regione di appartenenza.
- **Persistence:** vengono utilizzate diverse tecnologie in base alla tipologia di dato da registrare. Ci sono infatti dati di geolocalizzazione, di autenticazione, di traduzione e di componenti e per ognuno viene usato un servizio differente, ottimizzato per ogni aspetto e tipo di dato da re-

gistrare. Ad esempio, per i dati di autenticazione viene usato MySQL poiché usa un pattern relazionale e ha ottime performance e integrità di DB, per i dati geografici viene usato PostGIS poiché è tra i più utilizzati per questa tipologia di dati, e così via. Infine, per le comunicazioni tra i dispositivi IoT e il resto del sistema, viene utilizzato MosQuiTTo, un broker MQTT.

Il lavoro descritto in questo articolo si focalizza, oltre che alla definizione delle funzionalità dei servizi, su come essi comunicano tra di loro e sulla parte di persistenza dei dati, che ovviamente è molto importante se si vogliono ottimizzare le prestazioni. Questo ci permette infatti di capire come diverse tecnologie di storage e DB possono essere migliori di altri in base alla tipologia di dato da registrare in memoria. Nel caso di SweepTron, i dati sono banalmente salvati su file in prima battuta, ma vengono periodicamente inviati al server remoto che ospita un servizio di InfluxDB, un time series DB ottimizzato per effettuare storage di dati di serie temporali, che sono proprio i dati che vengono creati e salvati dalla nostra applicazione.

In entrambi gli esempi studiati, viene molto enfatizzata la proprietà dei microservizi riguardante la replicazione dei servizi. Infatti, questa è una delle più importanti proprietà, come già detto prima. Allo stesso modo, viene descritta esattamente la parte comunicativa dei servizi, fattore importante per il corretto funzionamento di qualsiasi sistema di questo tipo. Vedremo in questa tesi come i diversi servizi comunicano tra loro per eseguire le funzioni per cui sono stati definiti, in modo da evitare conflitti e al fine di risolvere problemi specifici dell'applicazione.

## 2.6 IQ Data e catture IQ

Una funzionalità fondamentale di SweepTron è quella di poter configurare le antenne installate sui nodi di acquisizione per effettuare delle particolari catture di segnali RF, chiamate catture IQ. Queste catture possono essere effettuate da alcune centraline Anritsu, come la MS27201 di cui si parlerà successivamente.

te. È importante specificare che, essendo un'attività particolarmente onerosa dal punto di vista computazionale, non tutti gli analizzatori sono in grado di effettuare questo tipo di cattura (e.g. l'analizzatore ultra-portable MS2760 usato per i nodi mobili di SweepTron non sono in grado di effettuare questo tipo di cattura).

Una cattura IQ, o acquisizione di dati IQ, si riferisce alla registrazione dei dati di segnale in formato "I" e "Q". Questi dati sono essenziali per l'analisi dei segnali RF e sono spesso utilizzati in applicazioni di monitoraggio dello spettro e analisi dei segnali.

Nel contesto dei segnali RF, il formato I/Q rappresenta due componenti fondamentali del segnale:

- I (In-Phase): Rappresenta la parte del segnale in fase con un riferimento di tempo.
- Q (Quadrature): Rappresenta la parte del segnale fuori fase rispetto al riferimento di tempo.

Catturare dati IQ significa registrare simultaneamente le ampiezze e le fasi del segnale in una data finestra temporale. Queste informazioni possono essere cruciali per analizzare la modulazione del segnale, identificare pattern, rilevare modelli di trasmissione e comprendere altri aspetti dettagliati del segnale RF. Questa tipologia di cattura può essere fatta solo da analizzatori di spettro avanzati, poiché la quantità di dati generata è molto grande, per cui il dispositivo deve avere delle capacità di computazione, elaborazione e storage opportune. Inoltre, la granularità della cattura, può essere configurata sullo strumento utilizzato tramite impostazioni come la risoluzione in bit del campionamento del segnale.

La Figura 2.5 rappresenta graficamente il contenuto di una cattura IQ tramite uno strumento di analisi IQ. Possiamo vedere in particolare due grafici che rappresentano la cattura:

- Costellazione: in questo grafico i dati IQ sono rappresentati come punti in uno spazio bidimensionale, dove l'asse delle ascisse rappresenta la

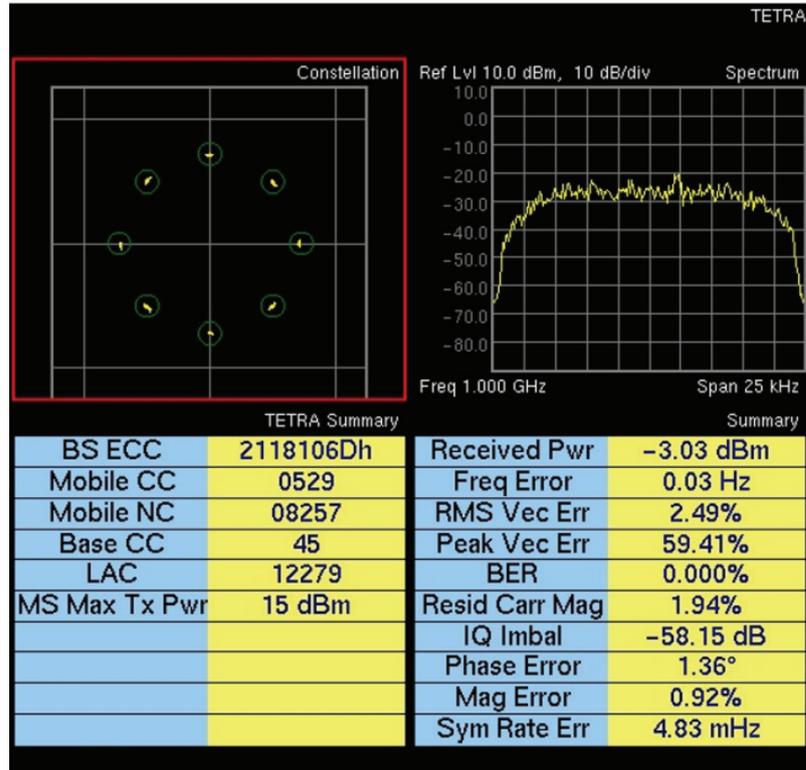


Figura 2.5: Esempio di cattura IQ

componente in fase (I) del segnale e l'asse delle ordinate rappresenta la componente in quadratura (Q) del segnale. Questa rappresentazione consente di visualizzare e interpretare il comportamento del segnale in termini di modulazione e rumore. Infatti, ogni punto nella costellazione rappresenta uno stato del segnale a un certo istante di tempo, e analizzando la forma e la distribuzione dei punti nella costellazione si possono ricavare informazioni importanti sul segnale, come la qualità della modulazione o la presenza di interferenze o disturbi.

- **Spettro:** in questo grafico viene mostrata la quantità di energia o potenza presente in ciascuna banda di frequenza all'interno di un intervallo di tempo specificato. Qui la potenza viene espressa in decibel-milliwatt (dBm). Questo tipo di rappresentazione fornisce informazioni cruciali sulla distribuzione della potenza del segnale RF rispetto alla frequenza, come la larghezza di banda del segnale o la scoperta delle frequenze portanti tramite lo studio dei picchi di potenza nello spettro.

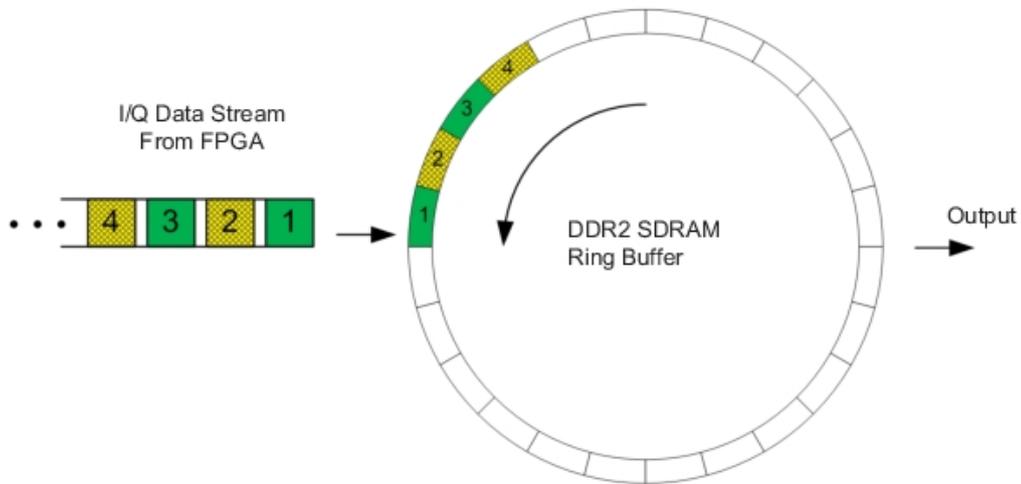


Figura 2.6: Funzionamento del buffer circolare per lo streaming di dati IQ

Oltre ai grafici, nella Figura 2.5 vediamo dei sommari che mostrano informazioni generali sulla cattura effettuata.

Una singola cattura IQ, in base a come viene configurata, può essere di dimensioni molto importanti (e.g. una cattura IQ lunga 1 secondo, con 16 bit di risoluzione e per una banda di 20 MHz può arrivare ad occupare anche uno o più GB di memoria). Per questo motivo, nel caso in cui volessimo acquisire dati IQ per poi inviarli a un PC esterno per analisi successive, effettuare il salvataggio del file di cattura IQ nella memoria secondaria dell'analizzatore può essere controproducente sia per motivi prestazionali (salvare una cattura IQ in locale nell'analizzatore, per poi inviare la cattura su un PC esterno causa una perdita di tempo evitabile), sia per motivi di occupazione di spazio (di solito lo spazio di storage degli analizzatori non è così elevata). Per questo motivo, è possibile effettuare, oltre all'acquisizione one-shot (o single) salvata sul dispositivo, anche una cattura in modalità streaming.

In questo tipo di cattura, considerando un analizzatore Anritsu di fascia alta (e.g MS27201A) che supporta le catture IQ, i dati IQ vengono acquisiti nella memoria SDRAM dell'analizzatore ad alta velocità, configurata come buffer circolare (come riportato in Figura 2.6). La velocità del flusso di dati verso la memoria è determinata dalla larghezza di banda IQ selezionata. Quando il buffer è pieno, i nuovi dati IQ vengono memorizzati nuovamente nel primo

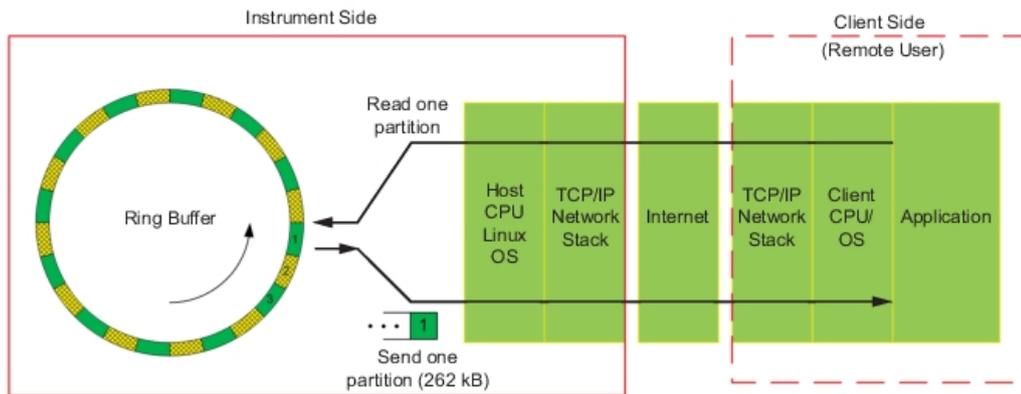


Figura 2.7: Streaming in remoto di una cattura IQ

slot del buffer, sovrascrivendo i dati precedenti.

In questo modo, dunque, è possibile ottenere dati IQ senza dover prima salvare i dati in un file, risparmiando in tempo e memoria del dispositivo di acquisizione. I dati vengono acquisiti con l'invio di un opportuno comando SCPI, che permette di ricevere dati relativi a una partizione del buffer circolare. Il formato dei dati ricevuti definisce un header della forma "#AXL\n", dove:

- A: è una singola cifra ASCII che specifica il numero di cifre in X.
- X: è un valore composto da una o più cifre ASCII che specificano il numero di byte di dati binari IQ e le coordinate ASCII della posizione GNSS (GPS).
- L: è la stringa ASCII contenente la posizione GNSS (GPS) in forma "latitudine, longitudine" in gradi decimali. Le coordinate registrano il punto in cui è stata attivata l'acquisizione IQ, se il modulo GPS è disponibile e se è stata trovata una posizione (e.g. se il dispositivo è coperto è possibile che il modulo GPS non riesca a calcolare una posizione).
- \n: segna la fine del componente di localizzazione GNSS (GPS) e l'inizio dei dati IQ.

Quindi, dopo il carattere di newline, avremo  $X - \text{len}(L)$  byte da leggere, relativi alla porzione di dati I/Q appena ricevuti.

Un'altra caratteristica importante dell'analizzatore è quella di poter inviare i dati relativi alle catture (anche IQ) in remoto, sempre per mezzo del buffer circolare. Possiamo vedere nella Figura 2.7 che l'analizzatore implementa lo stack TCP/IP, per cui la trasmissione via Internet dei dati è ammissibile. Tuttavia, per quanto riguarda SweepTron, questa possibilità non è stata esplorata e implementata poiché ci sono stati diversi casi di attacchi da parte di malintenzionati al fine di rendere inutilizzabili i dispositivi di acquisizione. Per questo motivo, nel nostro caso di studio, l'analizzatore è direttamente connesso al PC Embedded che lo controlla, al fine di evitare problemi di sicurezza.

## 3. SweepTron

In questo capitolo verrà descritto il sistema SweepTron, con tutte le componenti e tutte le funzionalità offerte.

### 3.1 Descrizione del sistema

L'obiettivo del sistema SweepTron è quello di effettuare monitoraggio dello spettro, acquisendo dati per diverse bande di frequenza configurabili, e offrire a questo proposito diverse funzionalità:

- Monitoraggio a durata flessibile: è possibile effettuare misurazioni "spot", oppure continue a lungo termine.
- Monitoraggio spaziotemporale: diversi nodi SweepTron possono effettuare misurazioni in luoghi e tempi diversi. I dati saranno poi immagazzinati in un database spaziotemporale per analisi future.

Nodi SweepTron dislocati nel territorio permettono di caratterizzare il segnale in una specifica area di copertura, oppure permettono di avere diversi dati ottenuti in diverse posizioni, città o Paesi.

- Algoritmi di misura e conversione per ottenere risultati in diverse forme (potenza del segnale in dBm, potenza dei campi in  $V/m$  o  $dBm/m^2$ ).
- Sistemi per garantire l'affidabilità ed il controllo remoto per avere resistenza ai guasti dovuti allo strumento di misurazione, riavvii improvvisi del sistema, blackout, ecc.

### 3. SWEEPTRON

<b>OS</b>	<ul style="list-style-type: none"> <li>Windows: i driver e il software delle centraline sono disponibili solo su Windows</li> </ul>
<b>Acquisizione dati</b>	<ul style="list-style-type: none"> <li>Acquisizione coppie timestamp - <math>dBm/m^2</math> e timestamp - <math>V/m</math> per analisi potenza di campo</li> <li>Acquisizione catture IQ</li> </ul>
<b>Processamento dati</b>	<ul style="list-style-type: none"> <li>Pre-processamento dati</li> <li>Compressione dati</li> </ul>
<b>Trasferimento dati</b>	<ul style="list-style-type: none"> <li>Invio coppie di dati al server remoto</li> <li>Invio catture IQ al server remoto</li> </ul>
<b>Salvataggio dati</b>	<ul style="list-style-type: none"> <li>Database influxDB spaziotemporale nel server remoto per dati di potenza di campo</li> <li>Salvataggio di file di catture IQ in memoria secondaria</li> </ul>
<b>Self-repair</b>	<ul style="list-style-type: none"> <li>Controllo dei servizi con riavvio in caso di crash o loop</li> <li>Startup all'avvio di Windows</li> </ul>

Tabella 3.1: Requisiti software del sistema SweepTron

La definizione e l'implementazione di SweepTron è iniziata a partire da un insieme di requisiti, riassunti nelle Tabelle 3.1, 3.2 e 3.3, e verranno discussi successivamente nello svolgimento della tesi.

In Figura 3.1 è mostrato uno schema riguardante il sistema SweepTron. Si può vedere che il sistema SweepTron prevede la definizione di due tipologie di nodi differenti:

- **Nodo fisso:** utilizza un analizzatore rack Anritsu MS27201A per effettuare misurazioni dello spettro. La frequenza massima di acquisizione è pari a 43.5 GHz. Inoltre, permette l'acquisizione di catture IQ, sia in modalità Single che Streaming. A questo analizzatore è collegata, via cavo coassiale low loss, un'antenna passiva Steatite Omni QMS-00017 le cui frequenze di acquisizione vanno da 0.8 a 40 GHz. L'analizzatore è poi collegato via cavo Ethernet (1 Gbps) a un mini PC embedded fanless AVS 520 (Windows 11, Intel Core i3-9100TE CPU @ 2.20GHz, 16 GB Ram, 240 GB SSD) che governa l'analizzatore tramite il software di SweepTron, processa e invia i dati al server remoto tramite un router con modulo SIM per connettività LTE.

### 3. SWEEPTRON

<b>Microservizi</b>	<ul style="list-style-type: none"> <li>• Servizi per acquisizione, processamento e trasferimento di dati, più meccanismi di self-repair</li> <li>• Coda di messaggi per comunicazione tra microservizi</li> </ul>
<b>Cron Job</b>	<ul style="list-style-type: none"> <li>• Avvio servizi di processamento e trasferimento di dati in specifici periodi della giornata</li> </ul>
<b>Allarmi</b>	<ul style="list-style-type: none"> <li>• Controllo messaggi di errore da centraline o servizi per richiedere intervento umano, nel caso in cui i meccanismi di self repair non siano sufficienti</li> </ul>
<b>Server remoto</b>	<ul style="list-style-type: none"> <li>• Implementazione di un microserver per il salvataggio dei dati ricevuti dai nodi e per la notifica degli allarmi</li> </ul>

Tabella 3.2: Scelte architetturali del sistema SweepTron

<b>MS2760A Ultra-portable</b>	<ul style="list-style-type: none"> <li>• Acquisizione soli dati relativi alla potenza di campo in dBm</li> <li>• Funzioni di conversione da dBm a <math>dBm/m^2</math> e da dBm a V/m con uso di antenna factor</li> </ul>
<b>MS27201A Rack</b>	<ul style="list-style-type: none"> <li>• Acquisizione dati di potenza di campo in <math>dBm/m^2</math> e catture IQ</li> <li>• Funzione di conversione da <math>dBm/m^2</math> a V/m</li> </ul>

Tabella 3.3: Requisiti di strumentazione del sistema SweepTron

- **Nodo mobile:** utilizza un analizzatore ultra-portable Anritsu MS2760A, di prestazioni più modeste. Infatti, la frequenza massima di acquisizione è pari a 32 GHz e non permette di effettuare catture IQ. L'antenna utilizzata è Anritsu 2000-1751-R le cui frequenze di acquisizione sono nei range 698-960/1710-2170/2500-2700 MHz. L'analizzatore è poi collegato via cavo USB 3.0 a un mini PC embedded fanless BPC 3022 (Windows 11, Intel Celeron N2930, 8 GB Ram DDR3, 128 GB SSD, modulo SIM per connettività LTE) che acquisisce i dati dall'ultra-portable tramite il software SweepTron, li processa e li invia al server remoto.

Un'altra differenza importante tra i due analizzatori descritti, è che, mentre l'analizzatore rack permette di acquisire dati sulla potenza dei segnali a diverse unità di misura, con conversioni effettuate dallo strumento stesso (e.g. dBm,  $dBm/m^2$ , V/m), l'analizzatore ultra-portable è in grado di effettuare acquisizioni solamente in dBm. Per questo motivo, nel caso in cui i dati da acquisire

### 3. SWEEPTRON

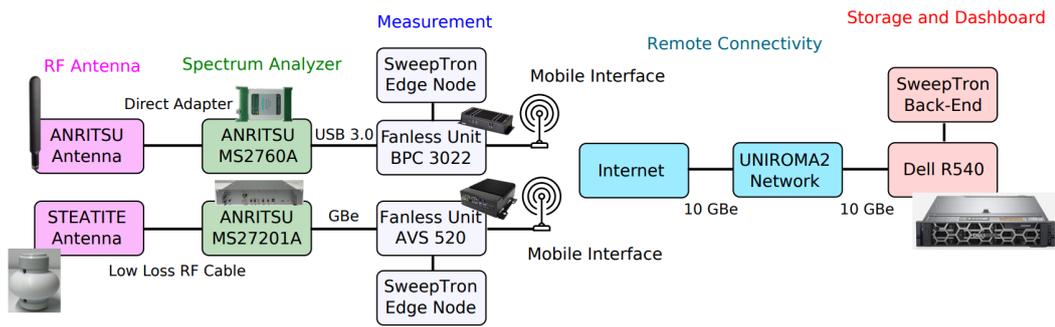


Figura 3.1: SweepTron E2E Breakdown



Figura 3.2: In (a) i componenti del nodo ultraportable: da sinistra, Analizzatore ultraportable Anritsu MS2760A, PC Embedded BPC-3022, Antenna Anritsu 2000-1751-r; in (b) i componenti del nodo fisso: da sinistra, Analizzatore rack Anritsu MS27201A, PC Embedded AVS520, Antenna Steatite Omni QMS-00017

siano in altre unità di misura, bisogna effettuare delle conversioni. SweepTron è in grado di convertire dBm in  $dBm/m^2$  oppure V/m, e le conversioni tra le diverse unità di misura sono le seguenti:

- La conversione da dBm a V/m è data dalla seguente formula:

$$V/M = \frac{1}{\sqrt{20}} * 10^{\frac{dBm+af}{20}}$$

dove  $af$  è detto Antenna Factor, e varia in base all'antenna scelta per l'acquisizione dei dati e in base alla banda di frequenza misurata. In particolare, i produttori di antenne forniscono un file CSV che contiene un fattore per ogni banda considerata.

- La conversione da  $dBm/m^2$  a V/m è data dalla seguente formula:

$$V/M = 10^{\frac{dBm/m^2+115.8-120}{20}}$$

dove il valore 115.8 deriva dalla seguente formula:

$$P = \frac{|E^2|}{Z_0}$$

dove  $Z_0$  rappresenta l'impedenza nello spazio libero, ed è pari a  $120\Omega$ .

Non è stato necessario effettuare altre conversioni per i nostri casi d'uso. Tuttavia, tornando al discorso dell'architettura a microservizi, è sufficiente estendere il servizio di sensing per inserire nuove conversioni utili per diversi casi e analisi.

Le componenti dei nodi sopra analizzati sono riportati in Figura 3.2. Queste sono le configurazioni definite per i nostri casi di studio, ma è ovvio che si possano scegliere altri PC, embedded e non, purché il sistema operativo utilizzato sia Windows e le caratteristiche hardware siano sufficientemente buone da mantenere lo stack software in esecuzione senza problemi. Allo stesso modo, potrebbe essere scelta un'antenna diversa, purché le bande di frequenza da monitorare siano coerenti con l'antenna scelta e con le capacità di acquisizione

dell'analizzatore. Nel nostro caso, la scelta di antenna e PC embedded dipende dalla tipologia di nodo e dal tipo di analisi che deve effettuare; infatti, per il nodo mobile è stato scelto un PC embedded con hardware meno performante rispetto al nodo fisso, e soprattutto con storage minore, poiché non permette di effettuare catture IQ, esegue meno microservizi e quindi ha un peso minore da un punto di vista di RAM e CPU.

Escludendo le differenze di capacità computazionali e funzionalità aggiuntive del nodo fisso, il funzionamento dei nodi è fondamentalmente lo stesso: il PC embedded esegue il software SweepTron che comunica con l'analizzatore, riceve i dati delle misurazioni e li salva in locale. Dopodiché, periodicamente, dopo un eventuale preprocessing dei dati, essi vengono inviati al server remoto. Quest'ultimo ospita un database spaziotemporale che immagazzina i dati per future analisi, e salva alcuni file in locale relativi alle catture IQ.

## 3.2 Descrizione del software

Il software eseguito dai nodi SweepTron è schematizzato nella Figura 3.3. Esso è composto da quattro microservizi diversi, che saranno analizzati in dettaglio nei paragrafi successivi. Il microservizio di sensing si occupa della comunicazione con l'analizzatore di spettro, e lo fa inviando comandi di configurazione dello strumento e richiedendo i valori delle misurazioni effettuate, oltre ad eventuali errori di configurazione o misurazione. Il microservizio di processing si occupa di effettuare pre-processamento di dati, in particolare andando a comprimere i dati relativi alle catture IQ, dato che essi possono avere una dimensione elevata. Il microservizio di transfer si occupa dell'invio in remoto delle misurazioni e delle catture IQ, oltre che dei log di errore. Infine, il microservizio di watchdog si occupa di riavviare i servizi nel caso in cui uno dei servizi vada in loop o in crash, e quindi non svolga più le sue funzioni ordinarie.

Per quanto riguarda il server remoto, esso implementa un server Flask [8]— un microframework web per Python open source e gratuito, che permette di

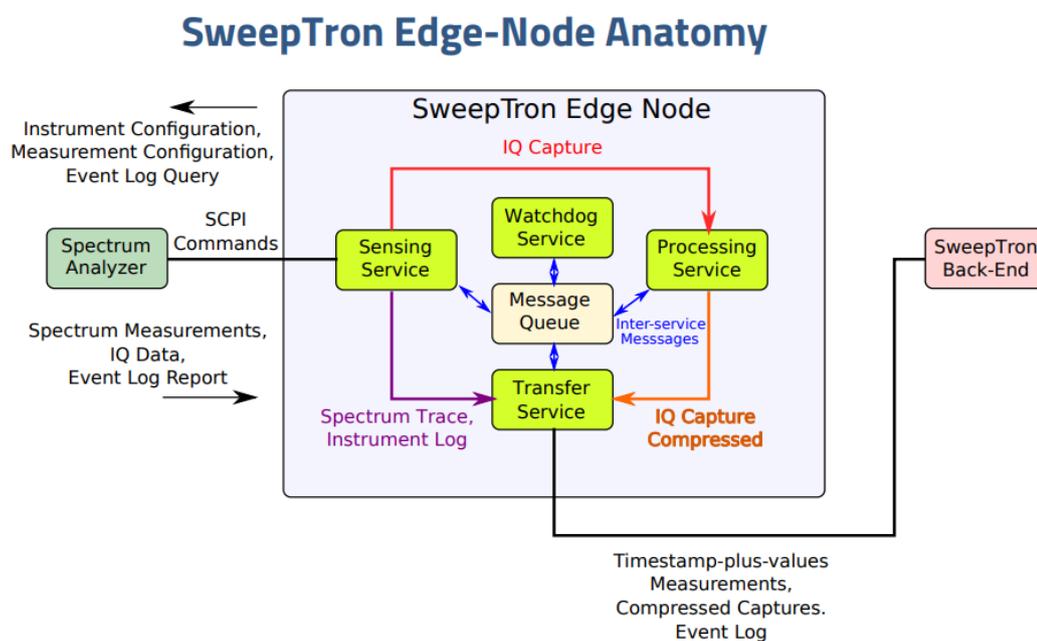


Figura 3.3: Struttura nodo SweepTron

creare rapidamente e facilmente applicazioni web in Python—che acquisisce i dati relativi alle misurazioni e li salva in un’istanza di database InfluxDB, oppure riceve file relativi a catture IQ o log e li salva in memoria secondaria.

I microservizi sopra descritti sono installati in Windows come servizi di sistema, e vengono lanciati all’avvio di Windows. L’avvio così impostato permette ai servizi di tornare in esecuzione anche in caso di riavvii dovuti ad aggiornamenti di sistema, blackout improvvisi, o altre cause.

### 3.2.1 Orchestrazione dei servizi

Analizzando il comportamento runtime di un nodo SweepTron si possono individuare due fasi di esecuzione:

- Fase di acquisizione: in questa fase sono attivi soltanto il microservizio di sensing e watchdog. Mentre il microservizio di sensing riceve dati dall’analizzatore e li salva in un file in locale, il microservizio di watchdog verifica che il sensing si comporti correttamente, attendendo ping periodici da parte di sensing.

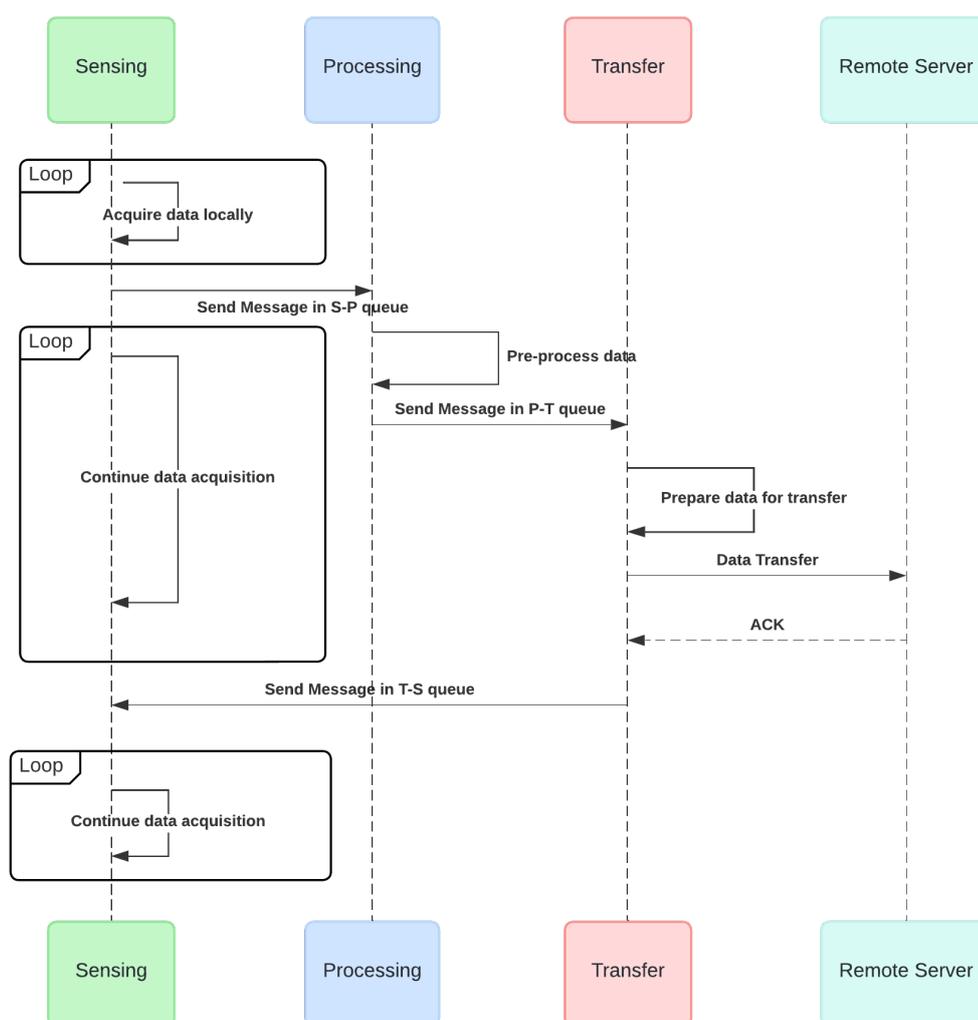


Figura 3.4: Scenario tipico di comunicazione in fase di trasmissione

- Fase di trasmissione: in questa fase tutti e quattro i servizi sono in esecuzione e, oltre all'acquisizione dei dati, viene effettuato anche il pre-processamento dei dati da parte del servizio di processing e il trasferimento dei dati al server remoto da parte del servizio di transfer. Il watchdog, in questa fase, controlla anche il corretto funzionamento dei servizi di processing e transfer, attendendo messaggi di ping periodici.

Per permettere il corretto funzionamento dei microservizi, in particolare da un punto di vista di comunicazione tra di essi, è necessario descrivere nei minimi dettagli il sistema di comunicazione. Possiamo vedere, nella Figura 3.4, un diagramma di sequenza che descrive lo scenario tipico di comunicazione

nella fase di trasmissione. In questo sistema, i servizi comunicano tra loro tramite un Message Broker che smista i messaggi in diverse code, in base agli attori della comunicazione. A questo proposito, vengono definite diverse code, una per ogni coppia mittente-destinatario di microservizi che devono comunicare tra loro. Dunque, le code che vengono definite ed utilizzate sono le seguenti:

- S-P: Usata dal microservizio di sensing per notificare al microservizio di processing che deve essere effettuato un pre-processamento di dati prima dell'invio al server remoto;
- S-T: Usata dal microservizio di sensing per notificare al microservizio di transfer che ci sono dei dati, da non pre-processare, che devono essere inviati al server remoto;
- S-W: Usata dal microservizio di sensing per notificare al microservizio di watchdog che esso è ancora in esecuzione normalmente;
- P-T: Usata dal microservizio di processing per notificare al microservizio di transfer che il pre-processamento è stato completato e quindi i dati processati possono essere inviati al server remoto;
- P-W: Usata dal microservizio di processing per notificare al microservizio di watchdog che esso è ancora in esecuzione normalmente, oppure per avvertirlo del suo spegnimento;
- T-S: Usata dal microservizio di transfer per notificare al microservizio di sensing che il trasferimento è stato completato;
- T-W: Usata dal microservizio di transfer per notificare al microservizio di watchdog che esso è ancora in esecuzione normalmente, oppure per avvertirlo del suo spegnimento;

Il message broker utilizzato attualmente è RabbitMQ [4], tra i più utilizzati attualmente per la sua interoperabilità con diversi protocolli standard, la sua flessibilità di configurazione e la sua affidabilità. Per la comunicazione

con il broker, SweepTron utilizza Pika [14], una libreria che fornisce un'implementazione in Python del protocollo AMQP (Advanced Message Queuing Protocol). AMQP è il protocollo di messaggistica utilizzato per la comunicazione tra applicazioni in SweepTron, e viene utilizzato al posto di MQTT perché, essendo l'applicazione totalmente in locale, non ha requisiti stringenti di rete, e quindi si possono sfruttare meglio i vantaggi di flessibilità di AMQP. In future evoluzioni di SweepTron in cui può essere studiata una fase di offloading e replicazione di servizi in diversi nodi, può essere interessante utilizzare anche MQTT e analizzare le differenze tra AMQP e MQTT da un punto di vista prestazionale.

Nei prossimi paragrafi verranno descritte le funzionalità offerte da ogni singolo microservizio, e le modalità di interazione con gli altri servizi per lo svolgimento delle attività principali di SweepTron.

#### 3.2.2 Sensing microservice

Il microservizio di Sensing è il più corposo dei quattro, poiché deve gestire tutte le comunicazioni con il dispositivo di misurazione. In particolare, questo servizio implementa funzioni di configurazione dello strumento e di acquisizione dati, oltre che ai meccanismi di conversione di dati descritti precedentemente. La comunicazione da/verso lo strumento di monitoraggio avviene tramite chiamate Standard Commands for Programmable Instruments (SCPI). Nel diagramma di flusso in Figura 3.5 è descritto il funzionamento di questo microservizio.

Per quanto riguarda la configurazione da un punto di vista di acquisizione di dati (quindi impostazioni dello strumento Anritsu), si possono impostare le seguenti caratteristiche :

- Modalità operativa: nel nostro caso si effettua analisi dello spettro di default, ma è possibile impostare altre configurazioni, come l'analisi dello spettro real time, misurazioni 4G/5G, e così via.

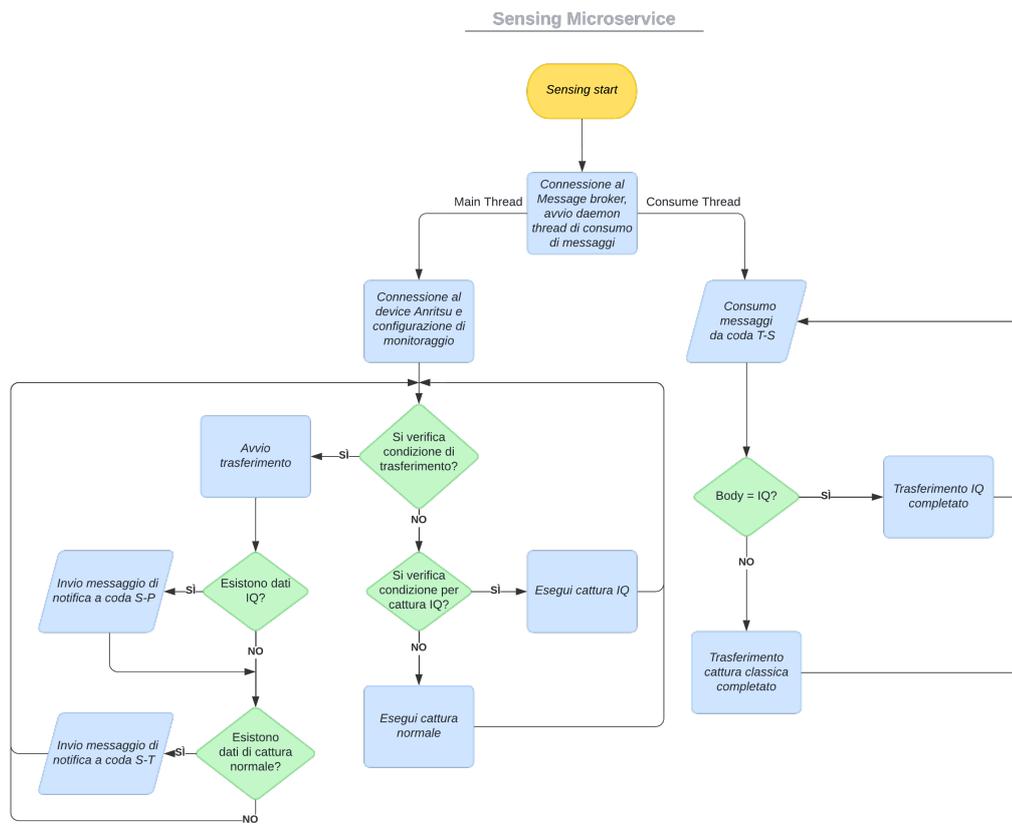


Figura 3.5: Flowchart microservizio di sensing

- Tipologia di misurazione: nel nostro caso si misura la potenza di canale, ma si può misurare ad esempio l'occupazione di banda. Questa è l'unica caratteristica tra quelle elencate che può essere impostata nell'analizzatore ultra-portable.
- Impostazione di misurazione della potenza di campo, con relativa scelta del fattore di antenna, necessario per una misurazione corretta del campo.
- Unità di misura della potenza di campo misurata (e.g.  $V/m$ ,  $dBm/m^2$ )
- Impostazioni iniziale di pre-amplificazione e guadagno.

Inoltre, per ogni banda monitorata, nel caso in cui si effettua una misurazione di potenza di campo:

- Frequenza iniziale e finale di banda da monitorare.
- Risoluzione di banda.
- Trace detection: nel nostro caso si acquisisce la potenza media (RMS) ma si può acquisire ad esempio il picco massimo o minimo.
- Trace type: specifica come vengono combinate le misurazioni effettuate (sweep) per ottenere i dati. Nel nostro caso si combinano con rolling average.

Se invece si effettua una cattura IQ, abbiamo parametri diversi:

- Risoluzione in bit della cattura;
- Durata della cattura;
- Modalità di cattura: singola o streaming;
- Formato fornito in output: ascii (per la rappresentazione human-readable di dati I e Q su più righe) o packed (che consiste in un insieme di byte ricevuta in una singola lettura);

- Sample rate e bandwidth della cattura;

Molte di queste opzioni possono variare in base al contesto. Ad esempio, le bande possono cambiare e possono essere assegnate a operatori telefonici differenti in base ad accordi commerciali (e.g. il band plan italiano è presente in Figura 4.1). Oppure, più semplicemente, si vuole avere la possibilità di tuning dei parametri di misurazione, o l'aggiunta di modalità di misurazione diverse in base al contesto di esecuzione, e così via. Per questo motivo, è stato definito un file di configurazione esterno per una impostazione semplificata dei parametri sopra descritti.

Solo elencando queste opzioni (che sono un piccolo sottoinsieme di quelle offerte dall'analizzatore rack) si comprende uno dei motivi per cui si utilizza un'architettura a microservizi: nel caso in cui si volesse effettuare un altro tipo di misurazione, oppure si volesse configurare un altro dispositivo di misurazione, è sufficiente estendere il microservizio di sensing, implementando le nuove tipologie di analisi e strumenti da supportare, lasciando invariati gli altri servizi. Se si fosse scelta un'architettura monolitica, l'aggiunta di nuove funzionalità avrebbe richiesto un rebuild totale dell'applicazione, senza considerare il rischio di complicare troppo il flusso di controllo dell'intera applicazione.

Una volta configurato l'analizzatore in modo che risponda alle esigenze di misurazione, i dati vengono acquisiti dal servizio e memorizzati in memoria secondaria all'interno di un file di log. A cadenza giornaliera (e.g. alle ore 4 di ogni giorno), si passa dalla fase di acquisizione alla fase di trasferimento, e quindi il servizio di sensing, nel caso in cui esistono dei dati da inviare al server remoto, invia dei messaggi alle code:

- S-P: avverte il servizio di processing nel caso in cui ci sono catture IQ salvate su file da inviare al server remoto.
- S-T: comunica al servizio di transfer che ci sono misurazioni di potenza di campo da inviare al server remoto, se ci sono.

In entrambi i casi, il servizio di sensing si pone in uno stato di trasferimento per il quale non effettua misurazioni a bande di frequenza uguali a quelle di invio,

finché tutti i dati non vengano trasferiti correttamente. Per questo motivo, mentre continua l'acquisizione per le altre bande di frequenza, si mette in ascolto in coda T-S per ricevere un messaggio di avvenuto trasferimento dei dati. Questo meccanismo verrà descritto più a fondo nei paragrafi successivi.

Prima di effettuare qualsiasi misurazione, il servizio di sensing invia un messaggio di ping alla coda S-W in modo da avvertire il servizio di Watchdog che esso sia correttamente in esecuzione. Considerando che viene effettuata un'acquisizione ogni 30 secondi circa, è stato impostato a 5 minuti il tempo dopo il quale il servizio viene riavviato dal watchdog, nel caso in cui il servizio non invii più messaggi alla coda S-W (e quindi, dal punto di vista del watchdog, risulta in loop o in crash, o comunque non sta seguendo il flusso di istruzioni programmato).

Ogni volta che il servizio di sensing effettua una chiamata SCPI verso l'analizzatore, potrebbero esserci degli errori causati da problemi di configurazione, oppure da disconnessioni del dispositivo. Nel primo caso, l'analizzatore può restituire un errore, che può essere recuperato tramite uno specifico comando SCPI. In questo modo, il microservizio recupera tutti i messaggi di errore inviati dall'analizzatore e li salva in un file di log, e nella fase di trasferimento gli errori salvati nel file di log vengono inviati al server remoto. In questo modo, è possibile segnalare se un qualche nodo sul campo è soggetto a errori di configurazione, e quindi intervenire prontamente per risolvere il problema e ristabilire il corretto funzionamento del nodo.

Inoltre, nel caso in cui si verifica un problema lato analizzatore, viene effettuata una reinizializzazione e riconfigurazione dello stesso, in modo che il nodo sia self-adaptive e possa autoripararsi in occorrenza di errori di configurazione.

#### **3.2.3 Processing microservice**

Il microservizio di processing si occupa di effettuare il pre-processamento dei dati acquisiti dal servizio di sensing. L'obiettivo del pre-processamento, in prima battuta, è quello di diminuire la dimensione dei dati da trasferire, senza

però ridurre troppo il grado di informazione degli stessi. Nel diagramma di flusso in Figura 3.6 è descritto il funzionamento di questo microservizio.

Questo microservizio in particolare deve essere avviato solo all'avvio della fase di trasferimento dei dati al server remoto, che come detto è a cadenza giornaliera. Per questo motivo, non è necessario che il servizio sia in esecuzione per l'intera giornata, bensì è sufficiente che sia in esecuzione nella finestra temporale di invio dei dati. Per modellare questo comportamento, un cron job governa l'avvio del microservizio di processing: in un momento specifico della giornata che coincide con l'inizio della fase di trasmissione, il cron job esegue il codice di avvio del servizio di processing che, dopo aver inviato un messaggio alla coda P-W per avvertire il processo di watchdog del suo avvio, attende il messaggio sulla coda S-P, lo consuma e inizia la fase di processamento.

Attualmente, il microservizio si occupa unicamente di comprimere i file relativi alle catture IQ utilizzando l'algoritmo Lempel-Ziv-Markov (LZMA). La scelta dell'algoritmo LZMA è dovuta al fatto che esso ha un elevato rapporto di compressione, quindi è perfetto per ridurre il più possibile la dimensione delle catture IQ, che per loro natura hanno dimensioni considerevoli.

Una volta pre-processati tutti i dati da processare (e.g. i dati della cattura IQ), il microservizio di processing invia un messaggio nella coda P-T per avvertire il microservizio di transfer che i dati preprocessati possono essere inviati al server remoto. Dopo un certo intervallo di tempo, il servizio di processing invia un messaggio di stop al microservizio di watchdog tramite la coda P-W per notificarlo del suo spegnimento. Dopodiché, esso termina la sua esecuzione.

Anche in questo caso, la scelta dell'architettura a microservizi permette di estendere facilmente questo servizio: ad esempio, si potrebbe scegliere di inviare anche i dati di cattura della potenza di campo, oppure si potrebbero effettuare medie temporali (con granularità al minuto, ora, giorno) in modo da ridurre ulteriormente i dati da inviare al server, oppure si potrebbe rinnovare l'algoritmo di compressione in futuro nel caso in cui ne venga sviluppato uno più performante, oppure usarne uno che sia computazionalmente meno oneroso nel caso in cui venga usato un PC embedded meno performante, e così via.

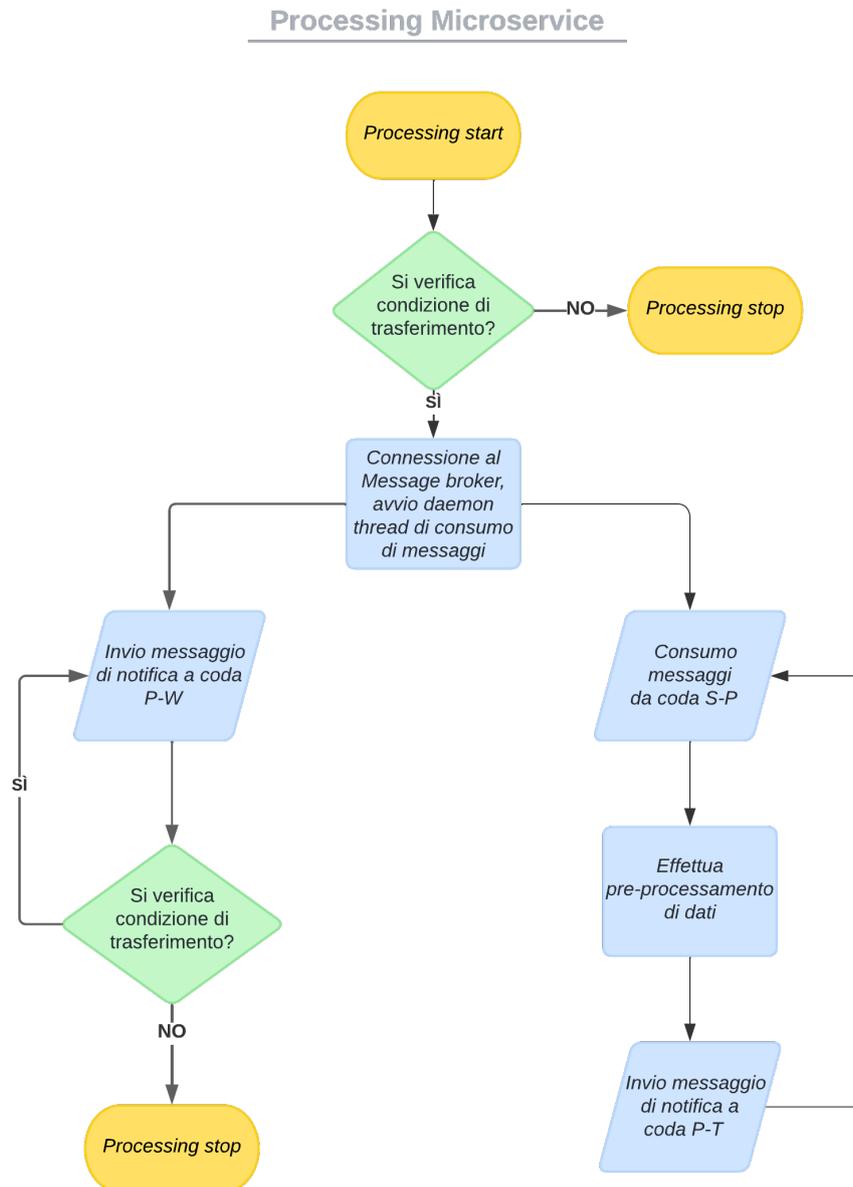


Figura 3.6: Flowchart microservizio di processing

### 3.2.4 Transfer microservice

Il microservizio di transfer si occupa del trasferimento dei dati acquisiti dall'analizzatore al server remoto. Nel diagramma di flusso in Figura 3.7 è descritto il funzionamento di questo microservizio.

Anche questo servizio, come nel caso del servizio di processing, viene avviato da un cron job al momento dell'inizio della fase di trasferimento. In particolare, in uno specifico momento del giorno, il cron job esegue il codice di avvio del servizio. Una volta in esecuzione, transfer invia un messaggio di ping nella coda T-W per avvertire il servizio di watchdog del suo avvio. A questo punto, si mette in ascolto di messaggi dalle code S-T e P-T. In base alla coda da cui riceve il messaggio può eseguire due funzioni di callback diverse:

- Messaggio da coda S-T: il servizio di transfer deve inviare al server i dati relativi alla cattura di EMF power. Questi dati sono salvati in un file di log. Per inviare questi dati, il servizio apre il file in lettura, legge il file e invia i dati al server riga per riga, lanciando tante richieste POST al server quante sono le righe da inviare. Ad ogni messaggio ricevuto, il server salverà i dati ricevuti nel database spaziotemporale, e risponde con una conferma di avvenuta ricezione.
- Messaggio da coda P-T: il servizio di transfer deve inviare al server i dati relativi alla cattura IQ. Questi dati sono stati pre-processati dal servizio di processing e sono salvati in uno specifico file. In questo caso, viene inviato al server l'intero file, poiché il server non deve salvare i dati nel db, bensì li salva in memoria secondaria. Questo perché i file di cattura IQ possono essere letti da programmi Anritsu specifici, per cui non ha senso salvarli su db. Infine, il server risponde con una conferma di avvenuta ricezione.

Terminato il trasferimento dei dati, in base al tipo di trasferimento terminato (dati EMF o cattura IQ) il servizio di transfer invia un messaggio nella coda T-S per comunicare al servizio di sensing che il trasferimento è terminato correttamente, in modo che sensing possa ricominciare ad acquisire normalmente

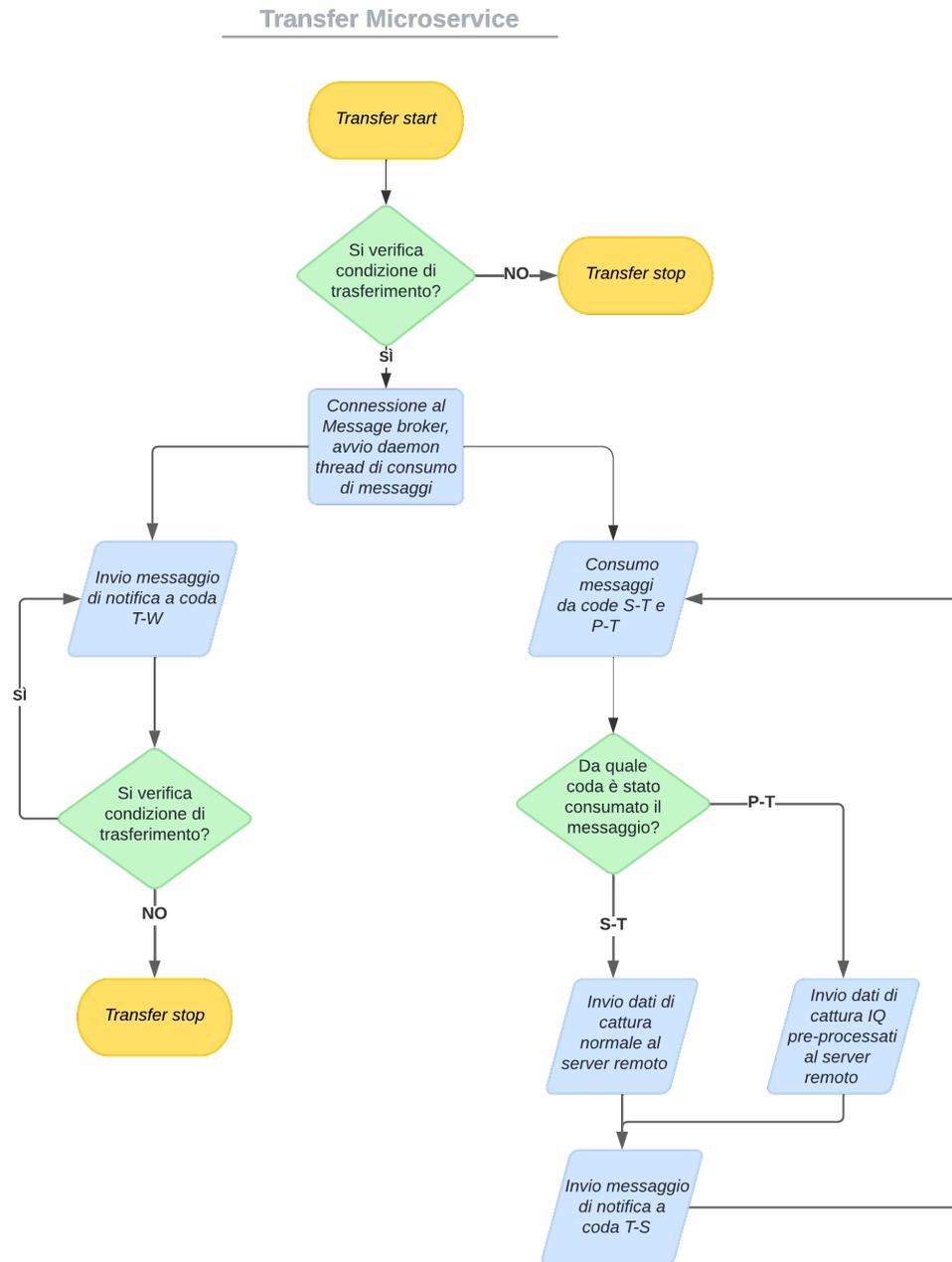


Figura 3.7: Flowchart microservizio di transfer

i dati a qualsiasi banda di frequenza. Inoltre, transfer invia un messaggio di stop al microservizio di watchdog tramite la coda T-W per notificarlo del suo spegnimento. Dopodiché, esso termina la sua esecuzione.

#### 3.2.5 Watchdog microservice

Il microservizio di watchdog si occupa di controllare che gli altri microservizi svolgano correttamente le loro attività. Nel diagramma di flusso in Figura 3.8 è descritto il funzionamento di questo microservizio.

In particolare, al suo avvio, esso si mette in ascolto di messaggi nelle code S-W, P-W, T-W. Sono stati definiti due tipologie di messaggi:

- Ping: notifica al watchdog che il servizio è in esecuzione. Se è la prima volta che un microservizio invia un ping, il watchdog mantiene in una variabile interna che quel microservizio è in esecuzione.
- Stop: notifica al watchdog che il servizio sta terminando, in modo che il watchdog non debba controllare che esso sia in esecuzione in modo corretto.

La logica adottata è la seguente: se il watchdog sa che un certo microservizio è in esecuzione, allora controlla costantemente che l'ultimo ping ricevuto non sia troppo datato. Nel caso in cui passi una quantità di tempo troppo elevata dall'ultimo ping ricevuto dal servizio, allora esso è considerato in loop o in crash e viene quindi avviata una procedura di restart del servizio. Il messaggio di stop è necessario perché in questo modo, se i microservizi di processing e transfer devono essere spenti perché l'esecuzione del nodo è in fase di acquisizione, allora il watchdog non deve controllarli, né tantomeno riavviarli.

Al fine di gestire questa logica, sono state definite tre variabili globali: *sensing\_activity*, *processing\_activity* e *transfer\_activity*. Queste rappresentano lo stato di attività dei tre microservizi controllati dal watchdog, e possono assumere i seguenti valori:

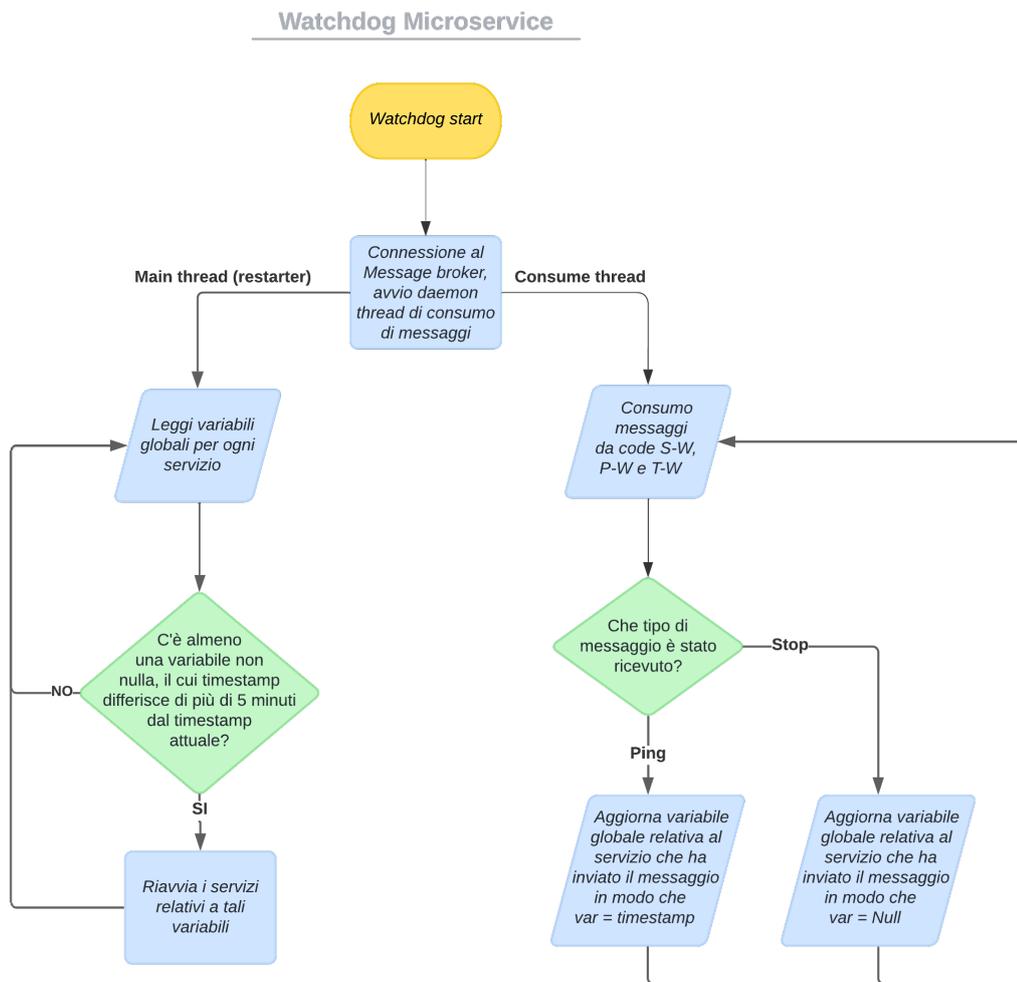


Figura 3.8: Flowchart microservizio di watchdog

- Null: se il servizio relativo alla variabile globale è spento. Viene impostato se il microservizio di watchdog riceve un messaggio di stop dal servizio controllato;
- Timestamp: se il servizio relativo alla variabile globale è in esecuzione. Viene impostato se il microservizio di watchdog riceve un messaggio di ping dal servizio controllato, e il valore della relativa variabile sarà pari al timestamp di ricezione del messaggio di ping;

Dunque, le variabili sopra descritte vengono controllate periodicamente, e se il loro valore è diverso da Null (il servizio non è spento) e la differenza tra il valore del timestamp della variabile e quello attuale è troppo alta, allora il servizio è considerato in loop o crash e va riavviato.

Attualmente, questo è l'unico collo di bottiglia dal punto di vista di resistenza a loop o crash: se il microservizio di watchdog dovesse comportarsi in modo anomalo per qualche motivo, si perderebbe la caratteristica di self-repair, e il watchdog andrebbe riavviato "a mano". Ad esempio, un malfunzionamento del servizio di watchdog potrebbe effettuare riavvii dei servizi di processing o transfer anche se SweepTron si trova in fase di acquisizione. Oppure, se il Watchdog andasse in crash per qualche motivo, gli altri tre microservizi non sarebbero più controllati e riavviati all'occorrenza. Per questo motivo potrebbero essere studiate altre forme di controllo, al fine di rendere il Watchdog più robusto.

### 3.3 Funzionalità aggiuntive

In questa sezione si descrivono le funzionalità aggiuntive gestite da SweepTron per il corretto funzionamento delle acquisizioni e le configurazioni lato analizzatore, oltre alla verifica del funzionamento dei microservizi.

### 3.3.1 File di log e funzioni di allarme

Come abbiamo detto, i microservizi definiti sopra sono costruiti e installati come servizi Windows. Per verificare che i servizi effettuino le loro operazioni in maniera consona al loro normale funzionamento, sono stati definiti quattro file di log su cui vengono salvate tutte le operazioni effettuate dai servizi in esecuzione. In questo modo, nel caso in cui ci siano malfunzionamenti o flussi di esecuzione errati da parte dei microservizi, i file di log possono essere consultati per identificare il problema e risolverli il prima possibile. Possiamo vedere un esempio di log di microservizio in Figura 3.9.

Oltre ai log dei microservizi, è ancora più utile definire un file di log che tenga conto degli errori o eccezioni lanciati dallo strumento di acquisizione. Infatti, le centraline Anritsu hanno la possibilità di restituire errore in caso di errori di configurazione, malfunzionamenti, surriscaldamento o sovratensione captata dallo strumento, e così via. Questi errori possono essere recuperati tramite il comando SCPA ":SYSTem:ERRor?". Nel caso di SweepTron, questi errori vengono salvati in un file di log apposito, e vengono periodicamente inviati al server remoto insieme ai dati relativi alle misurazioni effettuate dallo strumento. In questo modo, dal server remoto può essere notificato all'utente che c'è qualche malfunzionamento lato strumento, e si può intervenire in tal senso. Possiamo vedere un esempio di log relativo agli errori in Figura 3.10.

### 3.3.2 Gestione fase di trasferimento e controllo delle frequenze

L'obiettivo di SweepTron è quello di effettuare misurazioni di potenza di EMF per determinate bande di frequenza, impostate tramite un file di configurazione. Supponiamo di analizzare, tra le bande impostate, una banda di uplink 5G. Se ci troviamo in fase di trasferimento, il microservizio di sensing continua a comunicare con la centralina per l'acquisizione dei dati di potenza, mentre il microservizio di transfer inizia il trasferimento dei dati al server remoto. Tuttavia, se la banda di uplink utilizzata dal servizio di transfer coin-

### 3. SWEEPTRON

---

```
2024-03-27 11:05:01 - Device found in IP 10.0.0.2, port 9001
2024-03-27 11:05:01 - Inizio consume
2024-03-27 11:05:01 - Current Frequency: 3470.0, Starting time: 27-03-2024 11:05:01
2024-03-27 11:05:32 - Current Frequency: 3550.0, Starting time: 27-03-2024 11:05:32
2024-03-27 11:06:03 - Current Frequency: 3590.0, Starting time: 27-03-2024 11:06:03
2024-03-27 11:06:34 - Current Frequency: 3630.0, Starting time: 27-03-2024 11:06:34
2024-03-27 11:07:05 - Current Frequency: 3680.0, Starting time: 27-03-2024 11:07:05
2024-03-27 11:07:35 - Current Frequency: 3760.0, Starting time: 27-03-2024 11:07:35
2024-03-27 11:08:06 - Current Frequency: 763.0, Starting time: 27-03-2024 11:08:06
2024-03-27 11:08:37 - Current Frequency: 773.0, Starting time: 27-03-2024 11:08:37
2024-03-27 11:09:08 - Current Frequency: 783.0, Starting time: 27-03-2024 11:09:08
2024-03-27 11:09:39 - Current Frequency: 3470.0, Starting time: 27-03-2024 11:09:39
2024-03-27 11:10:10 - Current Frequency: 3550.0, Starting time: 27-03-2024 11:10:10
2024-03-27 11:10:41 - Current Frequency: 3590.0, Starting time: 27-03-2024 11:10:41
2024-03-27 11:11:11 - Current Frequency: 3630.0, Starting time: 27-03-2024 11:11:11
2024-03-27 11:11:42 - Current Frequency: 3680.0, Starting time: 27-03-2024 11:11:42
2024-03-27 11:12:13 - Current Frequency: 3760.0, Starting time: 27-03-2024 11:12:13
2024-03-27 11:12:43 - Current Frequency: 763.0, Starting time: 27-03-2024 11:12:43
2024-03-27 11:13:14 - Current Frequency: 773.0, Starting time: 27-03-2024 11:13:14
2024-03-27 11:13:45 - Current Frequency: 783.0, Starting time: 27-03-2024 11:13:45
2024-03-27 11:14:16 - Current Frequency: 3470.0, Starting time: 27-03-2024 11:14:16
2024-03-27 11:14:46 - Current Frequency: 3550.0, Starting time: 27-03-2024 11:14:46
```

Figura 3.9: Esempio di file di log. In questa figura trattasi di log del micro-servizio di Sensing.

```
04-04-2024 10:37:45 -141,"Invalid character data;character received: \n"
04-04-2024 10:37:45 -222,"Data Out of Range;IQ:BITS"
04-04-2024 10:37:45 -103,"Invalid separator;TRACE:IQ:DATA:FORM"
04-04-2024 10:37:45 -141,"Invalid character data;character received: \n"
04-04-2024 10:37:45 -141,"Invalid character data;character received: \n"
04-04-2024 10:37:45 -102,"Syntax error;SWEEP:MODE"
04-04-2024 10:37:45 -141,"Invalid character data;character received: \n"
```

Figura 3.10: Esempio di file di log relativo agli errori. Utilizzato per funzioni di allarme.

cide con la banda analizzata in quel momento dall'analizzatore, la misurazione può risultare sfalsata, poiché l'antenna usata dalla centralina può captare il segnale utilizzato dal PC embedded per l'invio dei dati.

Questa situazione deve essere evitata, e lo si fa in maniera semplice: all'interno del file di configurazione è possibile definire le bande di frequenza utilizzate nel trasferimento dei dati, che ovviamente dipende dalla SIM (e quindi dall'operatore) installata sul PC embedded. A questo punto, quando il servizio di sensing invia messaggi ai servizi di processing o transfer per iniziare la fase di trasferimento, si mette in stato di trasferimento, evitando di analizzare le bande di frequenza che, come definito nel file di configurazione, sono utilizzate per l'invio dei dati al server. Purtroppo non è possibile recuperare dinamicamente le bande di frequenza usate per il trasferimento.

Una volta terminato il trasferimento, il sensing riceve i messaggi di fine trasferimento da parte del servizio di transfer, e quindi ricomincia ad analizzare anche le bande utilizzate per l'invio dei dati in remoto.

## 4. Caso di studio: SweepTron TITAN monitoring node

L'argomento di tesi riguarda anche un caso di studio in cui un nodo fisso è stato installato a Bologna, Emilia Romagna, in particolare nell'edificio del Comune di Bologna. Questo nodo prende il nome di nodo SweepTron TITAN. La missione di installazione dell'apparecchiatura è stata svolta il giorno 7 marzo 2024, in collaborazione con il WiLab, il laboratorio nazionale delle comunicazioni wireless del Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT).

Il dispositivo è stato installato al decimo piano, e la scelta del luogo è ricaduta nel fatto che esiste copertura di segnale 5G per ogni banda utilizzata nel territorio italiano, sia sub-GHz che 3GHz, per ogni operatore. Le foto 4.2 rappresentano il nodo TITAN installato all'interno di un rack posto nell'edificio.

Possiamo vedere il piano italiano per le bande utilizzate nella Figura 4.1. Nel dispositivo configurato al comune di Bologna, si è scelto di effettuare misurazioni di potenza, oltre a catture IQ, solamente sulle bande 5G in downlink, quindi:

- Per le bande sub-GHz, si analizzano le bande n28 di Iliad, Telecom Italia e Vodafone (rispettivamente 758-768 MHz, 768-778 MHz, 778-788 MHz);
- Per le bande a 3.5GHz, si analizzano le bande n78 di Iliad, Vodafone, Telecom Italia (rispettivamente 3620-3640 MHz, 3640-3720 MHz, 3720-





(a)

Figura 4.2: Foto del nodo TITAN installato al Comune di Bologna. La centralina Anritsu MS27201A e il PC Embedded AVS 520 che esegue i servizi sono installati all'interno del rack in figura, e alimentati dall'edificio. Il PC Embedded è collegato a un router LTE con antenna posta all'interno del rack, mentre l'antenna STEATITE è posta al di fuori dell'edificio ed è collegata all'analizzatore rack tramite cavo coassiale, che passa attraverso un foro al muro che divide rack e antenna.

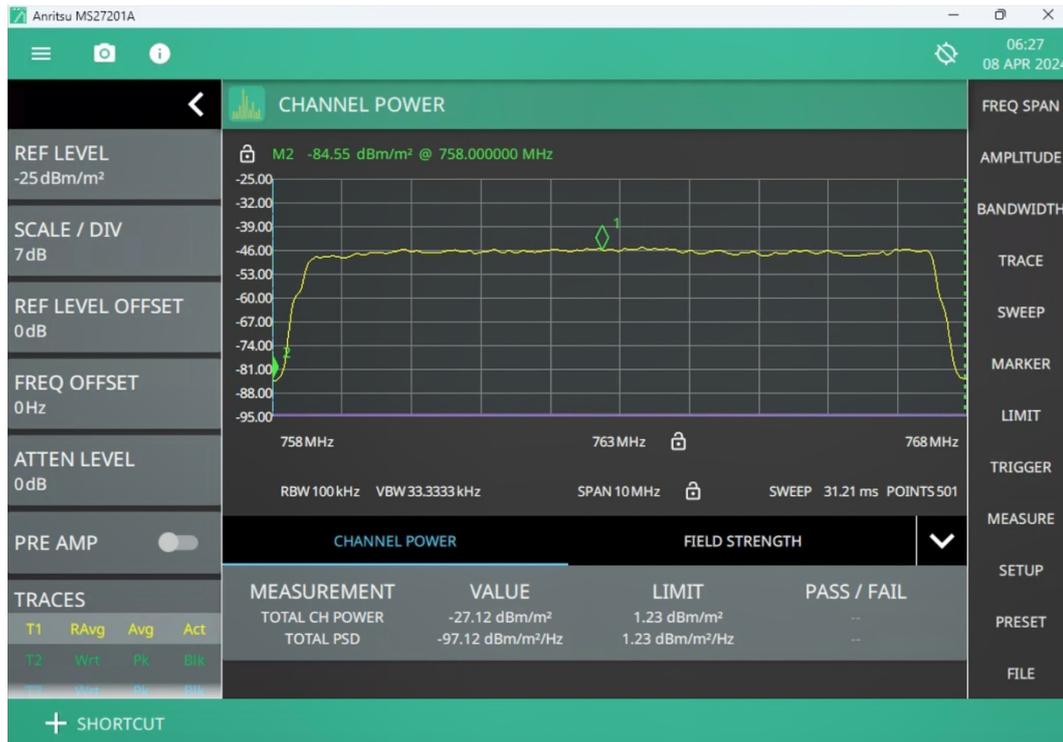


Figura 4.3: Software Anritsu MS27201A per Windows, in cui viene mostrata la configurazione della misurazione che si sta effettuando dal punto di vista di banda, unità di misura, finestra di visualizzazione, markers, e così via. Nel caso dell'immagine, si tratta del nodo TITAN, l'unità di misura dei dati acquisiti è  $dBm/m^2$  e la banda analizzata è la 758-768 MHz.

3800 MHz), oltre a bande regionali (3440-3500 MHz, 3540-3560 MHz, 3560-3620 MHz);

Il PC Embedded AVS 520 è stato preconfigurato, installando i quattro microservizi come servizi Windows e installando il servizio di RabbitMQ per la gestione della comunicazione. Inoltre, al fine di poter intervenire in remoto per l'eventuale modifica di parametri di configurazione, è stata installata una utility di desktop remoto per permettere l'utilizzo del dispositivo via Internet.

Da un punto di vista di configurazione generale, si è scelto di monitorare H24 le bande sopra descritte, e di effettuare una cattura IQ ogni ora per ogni frequenza. La fase di trasferimento inizia ogni giorno alle ore 4, e in questa fase i dati di potenza analizzati vengono inviati al DB InfluxDB in esecuzione nel server remoto, mentre i file relativi alle catture IQ vengono inviati e salvati nella memoria secondaria del server.

Nella Figura 4.3 possiamo vedere un esempio di configurazione dello strumento, grazie al software Anritsu relativo all'analizzatore rack. Un software analogo esiste per l'analizzatore ultraportable, ma come già detto permette una configurazione meno ampia, oltre all'impossibilità di configurare una connessione remota.

## 4.1 Risultati

Questo paragrafo descrive i risultati di misurazione ottenuti dal nodo TITAN per circa un mese di attività. In particolare, sono stati graficati i risultati ottenuti dal 9 al 30 marzo. Considerando il codice del servizio di sensing, l'analisi è effettuata per nove bande di frequenza, e in particolare i risultati ottenuti sono ottenuti in modo che, per ogni frequenza, vengono effettuate 40 misurazioni ogni 30 secondi, e le frequenze vengono scorse una alla volta (lo strumento non è in grado di effettuare misurazioni su più bande contemporaneamente, ma iterativamente analizza una banda per volta). Dunque, un ciclo di misurazione totale dura 4 minuti e mezzo, e per questo motivo si è deciso di effettuare smoothing dei dati ogni 5 minuti, in modo da mantenere più informazione possibile, dato che 22 giorni di dati sono relativamente pochi in questo caso per effettuare smoothing con granularità meno fine. In particolare, lo smoothing consiste nell'effettuare la media quadratica dei dati ottenuti ogni 5 minuti, in modo da avere un numero minore di dati graficati, e quindi un grafico più lineare e leggibile.

Dunque, per i dati ricevuti, è stata effettuata una fase di post-processing dei dati, divisa in questi punti:

1. Pulizia dei dati: sono stati rimossi dati misurati in maniera non corretta, come i valori "NaN";
2. Conversione dei valori ottenuti da  $dBm/m^2$  a lineare;
3. Smoothing come descritto sopra;
4. Conversione dei valori smoothed da lineare a  $dBm/m^2$ ;

Una volta effettuate queste operazioni, i dati sono pronti per essere graficati. In particolare, sono stati realizzati tre grafici che variano in base a come vengono cumulate le bande di frequenza misurate. In particolare, abbiamo i seguenti casi:

- Grafico dei risultati, banda per banda: otteniamo nove scatter plot che rappresentano l'andamento delle diverse bande misurate in maniera separata.
- Grafico dei risultati, cumulando le bande in base al gruppo di frequenze: in particolare, otteniamo due scatter plot, uno che riguarda l'andamento delle bande di frequenza sub-GHz (quindi la cumulativa delle bande 758-768 MHz, 768-778 MHz, 778-788 MHz) e uno che riguarda l'andamento delle bande di frequenza intorno a 3.5GHz (quindi la cumulativa delle bande 3440-3500 MHz, 3540-3560 MHz, 3560-3620 MHz, 3620-3640 MHz, 3640-3720 MHz, 3720-3800 MHz);
- Grafico dei risultati, cumulando le bande in base agli operatori telefonici: otteniamo qui cinque scatter plot per i diversi operatori (TIM, Vodafone, Wind-3, Iliad, Linkem), le cui bande sono definite in Figura 4.1;

Di seguito sono riportati i grafici sopra descritti. Essi mostrano valori in  $dBm/m^2$ , poiché i valori convertiti in  $V/m$  per alcune frequenze sono troppo piccoli per essere visualizzati in maniera leggibile. Nel caso di nodo TITAN non è possibile effettuare un confronto tra le utilizzazioni delle reti pre-5G e delle reti 5G, proprio perché l'analisi delle bande è ristretta alle sole reti 5G. Comunque, si possono confrontare le diverse bande e l'utilizzazione delle bande dei diversi operatori.

Nelle figure 4.4, 4.5, 4.6 possiamo innanzitutto notare come tutte le bande analizzate presentano una forte periodicità giorno/notte e giorno/settimana. Più in dettaglio, l'esposizione 5G è più bassa durante la notte, mentre i picchi sono concentrati nelle ore diurne. Questo comportamento è atteso, in quanto l'esposizione è influenzata dal numero di connessioni attive nell'area coperta (e ovviamente dalla quantità di traffico generato da ciascun terminale) i cui picchi

sono naturalmente localizzati quando la maggior parte delle persone vicine al comune di Bologna utilizzano la rete. Non si nota invece una periodicità giorno/settimana. In particolare, la differenza di esposizione in giorni feriali e festivi è minima.

Possiamo altresì notare che l'esposizione al 5G presenta picchi diversi, che si trovano sia durante i giorni lavorativi che nei fine settimana

Sempre in generale, non si riesce ad individuare nel tempo una crescita di utilizzazione del segnale 5G. Questo perché un periodo di misurazione di 20 giorni non è sufficiente a catturare questo andamento. Per questo motivo, non si intravedono trend secolari che ci aspettiamo dovrebbero essere di crescita nel caso di 5G in generale.

Focalizzando la discussione sul grafico in Figura 4.4, 4.5 notiamo due "gruppi" di bande: nel gruppo inferiore troviamo le bande a 773 MHz e 3550 MHz, e sono sottoutilizzate rispetto al gruppo superiore. In particolare, mentre dai dati relativi alla banda a 773 MHz si nota la periodicità giorno/notte discussi prima, indice quindi di una certa utilizzazione, seppur molto bassa, dai dati relativi alla banda a 3550 MHz non si evidenzia questo comportamento, anzi la linea del grafico è particolarmente stabile. Questo porta a pensare che la banda a 3550 MHz non sia utilizzata nell'area intorno al comune di Bologna in cui è stato installato il nodo TITAN.

Si nota in particolare la differenza tra i due gruppi nella Figura 4.5. Infatti, vediamo qui che le bande N78, quindi nell'intorno di 3.5GHz, sono ampiamente più utilizzate delle bande sub-GHz. Anche qui si nota l'andamento giorno-notte.

Infine, il grafico in Figura 4.6 ci permette di evidenziare il fatto che la banda inutilizzata discussa precedentemente è l'unica banda Wind-3 5G del territorio. Questo indica che non c'è copertura 5G per Wind-3 nella zona intorno al comune di Bologna. Inoltre, la banda TIM è poco utilizzata nella zona rispetto alle concorrenti Vodafone, Iliad e Linkem.

Continuando con l'acquisizione dei dati, acquisendo dati per periodi sufficientemente più grandi (da sei mesi a diversi anni) si potrà essere in grado

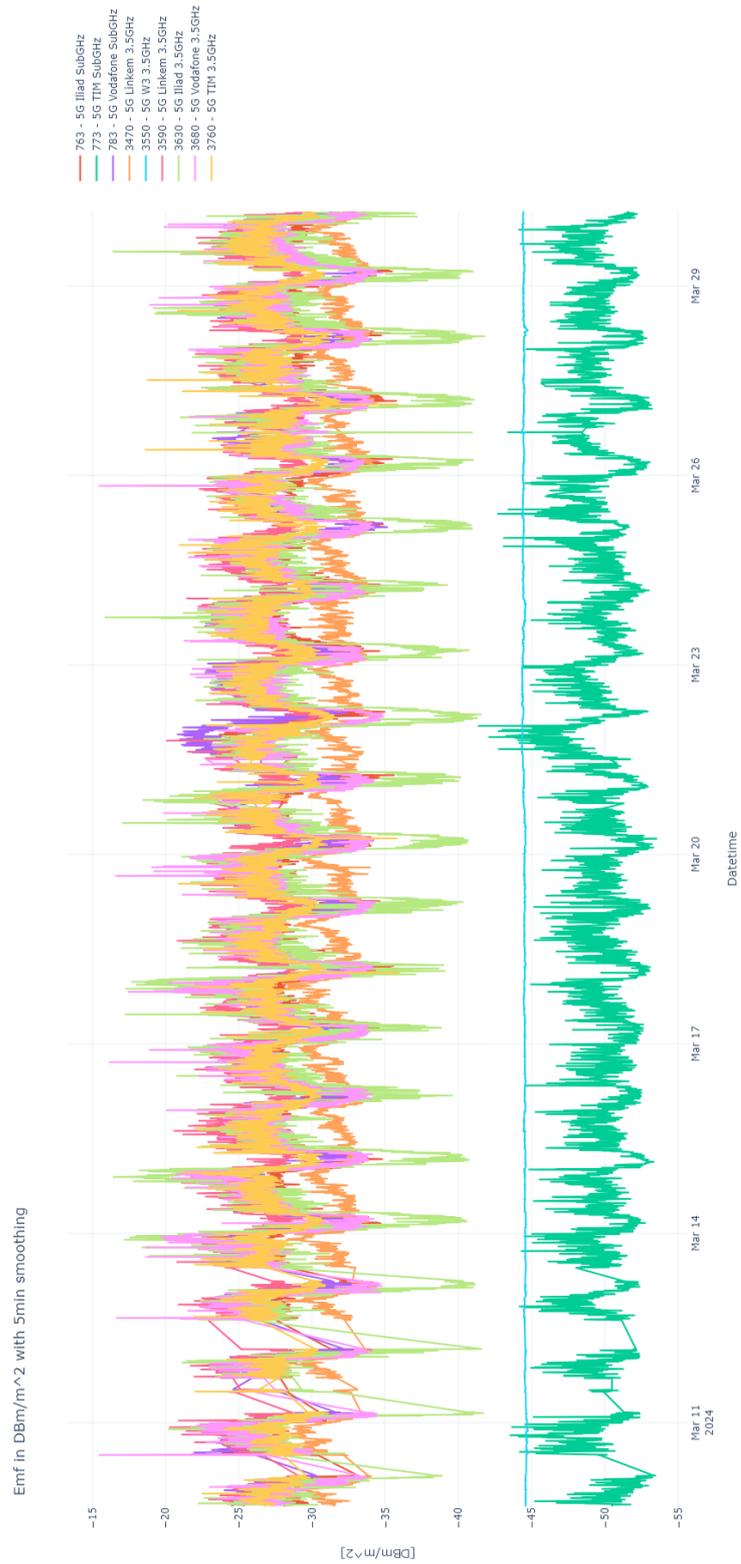


Figura 4.4: Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi nelle diverse bande in  $dBm/m^2$ .

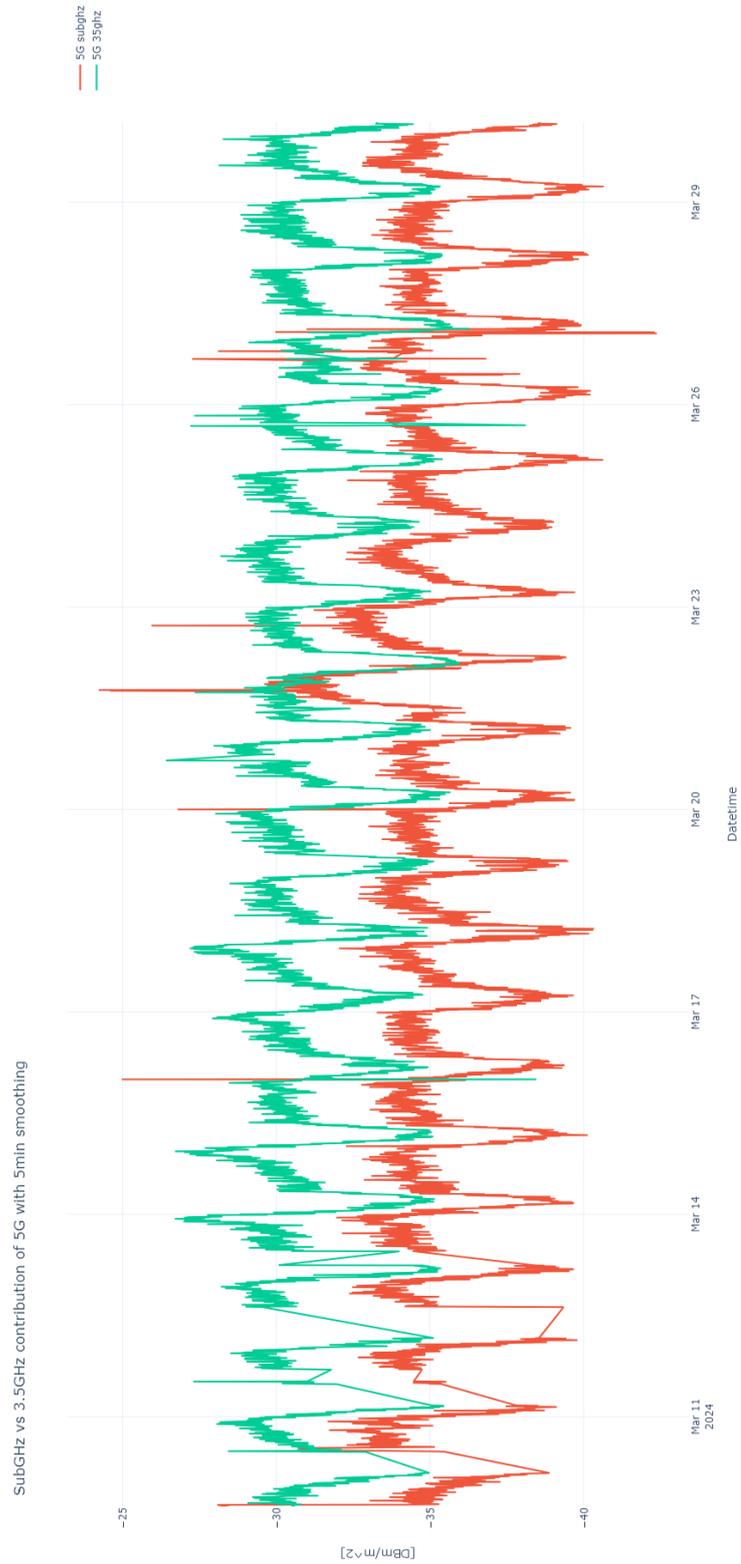


Figura 4.5: Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi, con comparazione per le bande subGHz e 3.5GHz, in  $dBm/m^2$ .

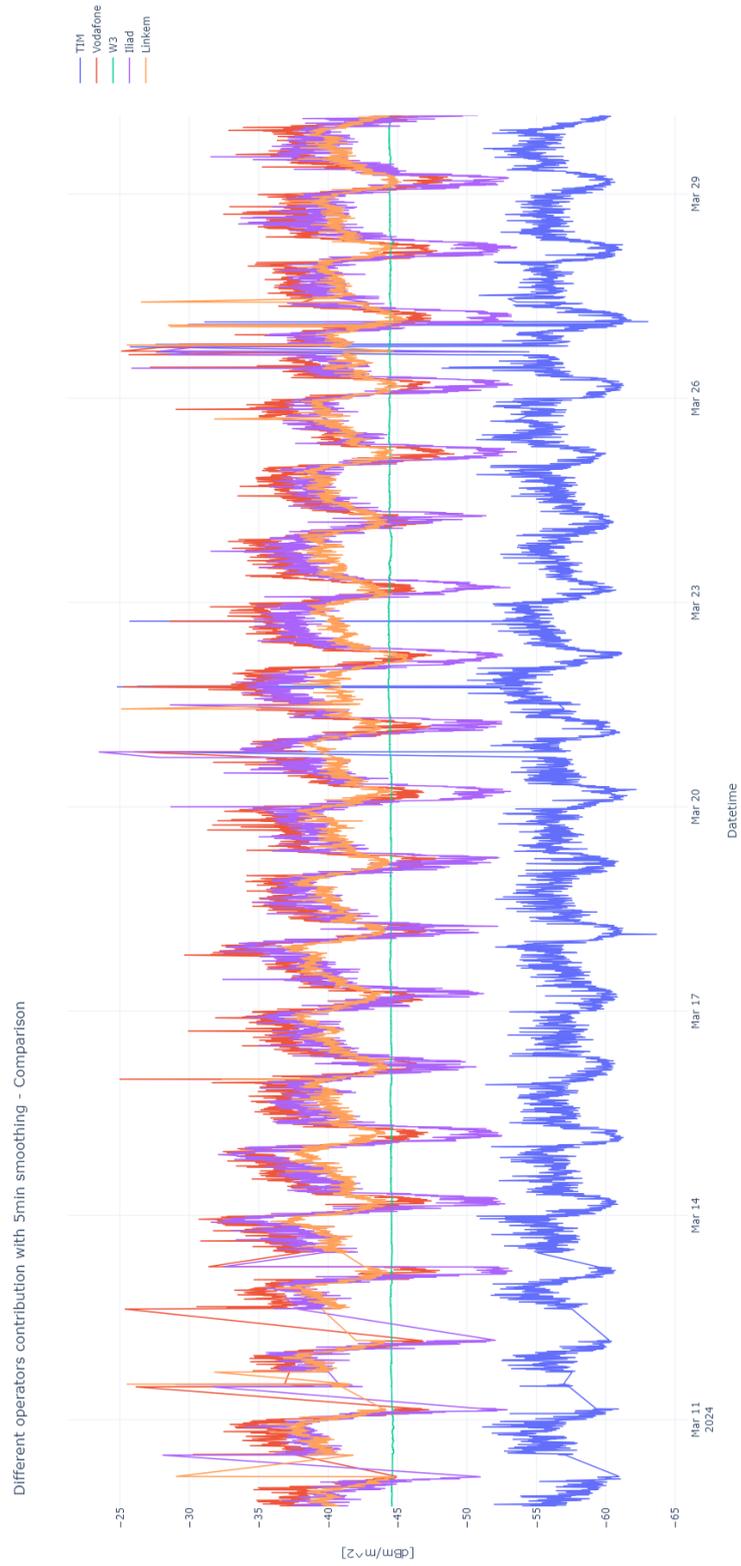


Figura 4.6: Grafico dei risultati di misurazione dal 9 al 30 marzo 2024, rappresentante la densità di potenza dei campi per i diversi operatori in  $dBm/m^2$ .

#### 4. CASO DI STUDIO: SWEEPTRON TITAN MONITORING NODE

di identificare trend secolari di crescita o decrescita dei segnali 5G. Potrebbe essere infatti interessante in futuro, soprattutto con l'avvento del 6G, vedere l'evoluzione, in termini di crescita o decrescita secolare, del segnale 5G in modo tale da poter anche comprendere e prevedere l'andamento delle bande con i salti generazionali futuri .

## 5. Conclusioni e studi futuri

Il sistema SweepTron nasce per creare un sistema di gestione di strumenti di misurazione (nel caso di studio specifico abbiamo visto analizzatori Anritsu), processamento e trasferimento dei dati che sia modulare, estensibile e ampiamente configurabile. L'architettura a microservizi, oltre a favorire le caratteristiche sopra citate, permette sia da un punto di vista di sviluppo di semplificare l'estensione dei servizi intervenendo direttamente sul microservizio da migliorare, sia da un punto di vista di scalabilità, che è il punto di forza principe dell'architettura a microservizi.

Per il momento, SweepTron non cavalca l'onda della scalabilità: sul nodo di misurazione vengono eseguiti tutti i microservizi di sensing, processing, transfer e watchdog, sia per motivi di necessità (nel caso di SweepTron TITAN è stata effettuata l'installazione di una singola macchina), sia per motivi di sicurezza (ci sono stati esempi recenti di centraline Anritsu bucate in remoto, quindi è stato necessario collegarlo localmente alla macchina via Ethernet); invece, il database spaziotemporale di InfluxDB è gestito da un servizio remoto, installato, se guardiamo il caso di studio, nel server di Roma Tor Vergata, ma in generale si potrebbe pensare di scalare anche lato server, per migliorare la sua disponibilità e la resistenza ai guasti su tutte.

Tuttavia, si potrebbero esplorare delle possibilità sulla via del cyber foraging e offloading dei servizi. Ad esempio, distribuire dei nodi addetti al solo trasferimento dei dati e introducendo una comunicazione tra servizi di transfer potrebbe ridurre il tempo atteso dal servizio di sensing per ricominciare ad analizzare le bande di trasferimento. Tuttavia, questi studi prevedono una fase di profiling dell'applicazione da un punto di vista di utilizzazione di CPU

e networking, per capire se può effettivamente avere senso utilizzare degli edge nodes in questo modo.

Un altro aspetto da considerare maggiormente può essere quello del self-repair: per il momento, il microservizio di Watchdog, insieme ai cron job, gestiscono l'avvio e il corretto svolgimento delle azioni degli altri microservizi. Tuttavia, se il Watchdog fosse soggetto a malfunzionamenti potrebbero esserci riavvii mancati o riavvii imprevisti dei servizi. Assicurare una maggiore robustezza da questo punto di vista potrebbe essere un altro argomento di studio.

# Bibliografia

- [1] Sam Aerts, Günter Vermeeren, Matthias Van den Bossche, Reza Aminzadeh, Leen Verloock, Arno Thielens, Philip Leroux, Johan Bergs, Bart Braem, Astrid Philippron, Luc Martens, and Wout Joseph. 2022. Lessons Learned from a Distributed RF-EMF Sensor Network. *Sensors* 22, 5 (2022). <https://doi.org/10.3390/s22051715>
- [2] Amazon. 2024. *What's the Difference Between Monolithic and Microservices Architecture?* Retrieved April 10, 2024 from <https://aws.amazon.com/compare/the-difference-between-monolithic-and-microservices-architecture/>
- [3] Atlassian. 2024. *Atlassian: Microservices vs. monolithic architecture.* Retrieved April 10, 2024 from <https://www.atlassian.com/it/microservices/microservices-architecture/microservices-vs-monolith>
- [4] Broadcom. 2007. *RabbitMQ: One broker to queue them all.* Retrieved April 10, 2024 from <https://www.rabbitmq.com>
- [5] Luca Chiaraviglio, Chiara Lodovisi, Daniele Franci, Settimio Pavoncello, and Tommaso Aureli. 2022. Six Months in the Life of a Cellular Tower: Is 5G Exposure Higher than pre-5G One?. In *2022 IEEE International Symposium on Measurements & Networking (M&N)*. 1–6. <https://doi.org/10.1109/MN55117.2022.9887670>
- [6] Nikola Djuric, Nikola Kavecán, Dragan Kljajic, Gorana Mijatovic, and Snezana Djuric. 2019. Data Acquisition in Narda's Wireless Stations based EMF RATEL Monitoring Network. In *2019 International Conference on Sensing and Instru-*

- mentation in *IoT Era (ISSI)*. 1–6. <https://doi.org/10.1109/ISSI47111.2019.9043671>
- [7] Nikola Djuric, Dragan Kljajic, Teodora Gavrilov, Vidak Otasevic, and Snezana Djuric. 2022. The EMF Exposure Monitoring in Cellular Networks by Serbian EMF RATEL System. In *2022 IEEE International Symposium on Measurements & Networking (M&N)*. 1–6. <https://doi.org/10.1109/MN55117.2022.9887716>
- [8] Miguel Grinberg. 2018. *Flask web development*. " O'Reilly Media, Inc."
- [9] Icnirp. 2020. Guidelines for Limiting Exposure to Electromagnetic Fields (100 kHz to 300 GHz). *Health Physics* 118, 5 (5 2020), 483–524. <https://doi.org/10.1097/hp.0000000000001210>
- [10] ICNIRP. 2024. *The International Commission for Non-Ionizing Radiation Protection (ICNIRP)*. Retrieved April 10, 2024 from <https://www.icnirp.org/>
- [11] Thomas Hunter II. 2017. *Advanced Microservices*. Apress.
- [12] MCA. 2024. *Malta Communications Authority (MCA) - Spectrum Monitoring*. Retrieved April 10, 2024 from <https://www.mca.org.mt/initiatives/spectrum-monitoring>
- [13] Manel Mena, Antonio Corral, Luis Iribarne, and Javier Criado. 2019. A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things. *IEEE Access* 7 (2019), 104577–104590. <https://doi.org/10.1109/ACCESS.2019.2932196>
- [14] pika. 2009. *GitHub - pika/pika: Pure Python RabbitMQ/AMQP 0-9-1 client library*. Retrieved April 10, 2024 from <https://github.com/pika/pika>
- [15] RATEL. 2024. *RATEL EMF Monitoring System*. Retrieved April 10, 2024 from <https://emf.ratel.rs>
- [16] RATEL. 2024. *RATEL: Regulatory Agency for Electronic Communications and Postal Services*. Retrieved April 10, 2024 from <https://www.ratel.rs/>

- [17] Roberto Sánchez-Reolid, Daniel Sánchez-Reolid, Clara Ayora, José Luis de la Vara, António Pereira, and Antonio Fernández-Caballero. 2023. Generic Architecture for Multisource Physiological Signal Acquisition, Processing and Classification Based on Microservices. In *Distributed Computing and Artificial Intelligence, 20th International Conference*, Sascha Ossowski, Pawel Sitek, Cesar Analide, Goreti Marreiros, Pablo Chamoso, and Sara Rodríguez (Eds.). Springer Nature Switzerland, Cham, 123–133.
- [18] Roberto Sánchez-Reolid, Daniel Sánchez-Reolid, António Pereira, and Antonio Fernández-Caballero. 2023. Acquisition and Synchronisation of Multi-source Physiological Data Using Microservices and Event-Driven Architecture. In *Ambient Intelligence—Software and Applications—13th International Symposium on Ambient Intelligence*, Vicente Julián, João Carneiro, Ricardo S. Alonso, Pablo Chamoso, and Paulo Novais (Eds.). Springer International Publishing, Cham, 13–23.
- [19] Francesco Troisi, Marina Boumis, and Paolo Grazioso. 2008. The Italian national electromagnetic field monitoring network. *annals of telecommunications - annales des télécommunications* 63, 1 (2008), 97–108. <https://doi.org/10.1007/s12243-007-0011-4>

# Ringraziamenti

Alla fine, anche io sono arrivato al momento dei ringraziamenti! Non posso che ringraziare i professori Alessandro Pellegrini e Luca Chiaraviglio per l'aiuto e il supporto che mi hanno dato per lo svolgimento di questa tesi: dall'inizio della progettazione all'esperienza della missione di Bologna, fino alla stesura della tesi, mi avete sempre accompagnato con professionalità ma anche con umanità e simpatia, e per questo vi ringrazio.

Ringrazio i miei amici "The Gentleman", che mi hanno dato modo di evadere dalle fatiche giornaliere con passeggiate (a Sora!), chiacchierate, allenamenti e tanto altro. Con voi c'è sempre modo di divertirsi, tra morra, memes e bevute, abbiamo passato anni fantastici, che non dimenticherò mai. Ringrazio anche gli amici dei "Ganjalfiani" (che tutto fanno, tranne ciò che direbbe il nome) per la compagnia, i videogiochi (la mia più grande passione), le serie, gli anime, e le cene al sushi! Un ringraziamento va anche ai miei amici dell'università, dai più "antichi" ai più recenti. Mi avete aiutato molto, vi ho stressato all'infinito in certi momenti, e non vi ringrazierò mai abbastanza per il vostro supporto.

Ringrazio la mia fidanzata Denise, che mi ha accompagnato negli ultimi 8 anni della mia vita, mi ha supportato e sopportato (soprattutto negli ultimi mesi) e mi è stata sempre vicina, nei momenti di sorrisi e nei momenti di pianto. Ne hai dovute sopportare tante (se continuavo ancora mi prendevi a sberle), e per questo ti ringrazio.

Infine, un ringraziamento speciale va alla mia famiglia: papà Domenico e mamma Pasqualina, che hanno lavorato e lavorano ogni giorno e senza di loro non sarei

## 5. RINGRAZIAMENTI

---

arrivato a questo punto, e mio fratello Piero, che nonostante la distanza geografica siamo sempre molto vicini. Loro mi hanno dato la forza di arrivare fino ad ora, e me ne daranno ancora per continuare a crescere e raggiungere i miei obiettivi.

VAAAAAAIIIIIIII!!!