DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA E TECNOLOGIE PER LA PRODUZIONE DEL SOFTWARE

TESI DI LAUREA
IN
INGEGNERIA DEL SOFTWARE

ORCHESTRAZIONE DI UN DOCKER SWARM PER IL MONITORAGGIO E L'ANALISI STAGIONALE DEI DISPLACEMENT DEI PERSISTENT SCATTERERS

RELATORE:
Chiar.mo Prof. Filippo Lanubile

LAUREANDO:
Julian Pajo

ANNO ACCADEMICO 2023/2024

Sommario

Abstract 4

Introduzione 5

1. Telerilevamento radar da Satellite 8

1.1 Funzionamento del Synthetic Aperture Radar (SAR) 9

1.2 Onde elettromagnetiche 11

1.3 Immagini SAR 16

2. Interferometria e Persistent Scatterers 18

2.1 InSAR: Interferometria Radar 18

2.2 PSInSAR: Persistent Scatterer Interferometric Synthetic Aperture Radar 20

2.2.1 Creazione della Serie Temporale del Displacement 21

2.2.2 Formazione dell'Interferogramma SAR 24

2.3 Analisi della Time Series del Displacement dei PS 26

2.3.1 Costruzione del Modello di Deformazione 26

3. Architettura del sistema 28

3.1 Docker 29

3.1.1 Concetti base 29

3.1.2 Architettura di Docker 31

3.1.3 Docker Swarm: deployment e orchestrazione 32

3.1.4 Nodi in Docker Swarm: Differenze tra Manager e Worker 33

3.1.5 Docker Networks 35

3.2 Componenti del Sistema 36

3.2.1 Development Environment (stack-dev) 37

3.2.2 Authentication (stack-auth) 40

3.2.3 Traefik (stack-traefik) 43

3.3 Illustrazione delle Interazioni tra i Diversi Componenti 45
3.3.1 Raccolta dei Dati 45
3.3.2 Autenticazione e Autorizzazione 46
3.3.3 Elaborazione e Visualizzazione dei Dati 49
4. Implementazione della Soluzione: EULER 51
4.1 Implementazione Pratica del Sistema 51
4.1.1 Ambiente di Sviluppo 52
4.1.2 Dataset 53
4.1.3 Configurazione del Docker Swarm 54
4.1.4 Sviluppo dei Microservizi 56
4.1.5 Deployment e Gestione delle Versioni 67
5. Applicazioni e prospettive future 69
Conclusioni 71
Bibliografia 73

Abstract

L’accuratezza nel monitoraggio della deformazione del suolo tramite i Persistent Scatterers (PS) è fondamentale per la comprensione dei processi geofisici. Tuttavia, le misurazioni possono essere influenzate da variazioni stagionali i quali effetti possono causare movimenti del suolo dovuti a cambi di temperatura o crescita della vegetazione.

Lo studio condotto in questa tesi si concentra sulla definizione di una soluzione tramite l’uso di un Docker Swarm per orchestrare un sistema di microservizi che monitorino i PS e filtrino le fluttuazioni periodiche, migliorando così l’accuratezza e l’affidabilità dei dati generati dal processo di Persistent Scatterer Interferometry (PSI). L’approccio mira a isolare il dato grezzo dalle distorsioni stagionali, fornendo una base più solida per l’analisi dei fenomeni di spostamento al suolo.

Nel primo capitolo di questo lavoro, si discuterà del telerilavamento in generale, con un focus particolare su quello satellitare. Verranno osservate le differenze tra le bande spettrali utilizzate dai vari sensori, per poi soffermarsi sui sensori radar e sulla tecnologia SAR (Synthetic Aperture Radar).

Il secondo capitolo sarà dedicato all'impiego della tecnologia SAR per la rilevazione millimetrica degli spostamenti del suolo. Verrà introdotto il concetto di Persister Scatterer, mettendo in luce i suoi vantaggi e le sue limitazioni.

Nel terzo capitolo si discuterà sull’architettura del sistema proposto, fornendo una panoramica dettagliata dei principali componenti del sistema e delle loro interazioni. Si parlerà delle decisioni progettuali che hanno portato all’adozione di Docker Swarm come framework di orchestrazione.

Nel quarto capitolo si discuterà dell’ implementazione di EULER: il sistema di monitoraggio dei displacement dei Persistent Scatterers (PS). Verrà dettagliato il processo di implementazione pratica del sistema, con focus sui passaggi per configurare Docker Swarm e sviluppare i microservizi necessari.

Nel quinto capitolo si osserveranno le diverse possibili applicazioni del sistema e le prospettive future.

Introduzione

Lo sviluppo umano e i cambiamenti naturali hanno determinato trasformazioni rilevanti nel paesaggio globale, specialmente negli ultimi secoli. Le pratiche di utilizzo del suolo, insieme a eventi naturali come frane, inondazioni e variazioni climatiche, hanno condotto a diverse forme di degrado ambientale. Di conseguenza, è diventato essenziale monitorare questi cambiamenti. L'approccio più efficace per realizzare ciò è naturalmente dall'alto.

La soluzione a questa necessità è stata fornita dal remote sensing.

Il remote sensing, o telerilevamento, ha rivoluzionato il modo in cui osserviamo e comprendiamo il nostro pianeta. Non bisogna limitarsi a collegare il remote sensing solo al monitoraggio satellitare, poiché questa tecnologia ha origini molto più antiche.

Figura 1: 1903 pigeons wearing cameras. Image Credit: NASA

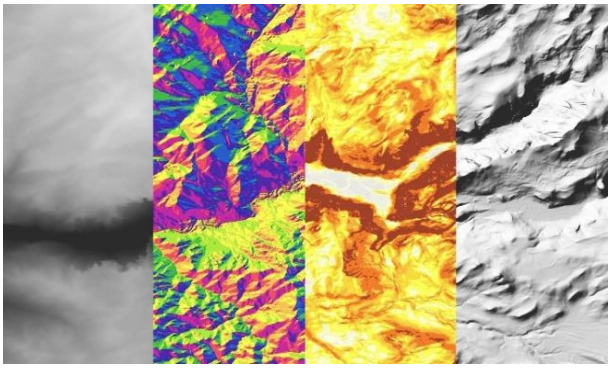
Infatti, il remote sensing, secondo la definizione più generale, è l'acquisizione di informazioni su un oggetto o un fenomeno senza dover essere in contatto diretto con esso.

Interpretando il termine in modo moderno, possiamo dire che i primi esperimenti di telerilevamento risalgono agli anni '40 del XIX secolo, quando furono scattate le prime fotografie da mongolfiere. All'inizio del XX secolo, invece, una delle idee più innovative consisteva nel fissare fotocamere su piccioni (Figura 1). Questi, una volta liberati, erano in grado di scattare fotografie dall'alto delle città.

Da quel momento, il telerilevamento ha visto una rapida evoluzione tecnologica. Si è passati dalle orto-foto aeree alle prime fotocamere installate su capsule spaziali, fino ad arrivare agli odierni satelliti progettati e lanciati in orbita appositamente per osservare la Terra e monitorare le attività umane dallo spazio.

Ad oggi sono tantissimi gli scopi e gli ambiti di ricerca per i quali è applicato il remote sensing: meteorologia, oceanografia, studio del cambiamento climatico, monitoraggio del suolo, delle attività antropiche, delle foreste e delle risorse idriche e altre.

(a) Vista da satellite di una città e delle campagne limitrofe, nel vicino infrarosso. Image Credit: ESRI



(b) Vista da satellite di una montagna, in più bande spettrali. Image Credit: ESRI

Figura 2: Alcuni esempi di immagini satellitari.

Esistono numerosi tipi di sensori utilizzati nel telerilevamento, ciascuno sensibile a diverse bande dello spettro elettromagnetico. Questi sensori si dividono principalmente in due categorie: sensori passivi e sensori attivi. I sensori passivi registrano le onde elettromagnetiche emesse da fonti esterne, come il Sole, mentre i sensori attivi emettono attivamente onde elettromagnetiche e registrano il segnale riflesso.

Tra i sensori attivi, uno dei più diffusi è il radar (Radio Detection And Ranging). Questo tipo di sensore utilizza onde radio per illuminare la superficie terrestre e rilevare il segnale riflesso.

Grazie alla sua capacità di operare in diverse condizioni atmosferiche e di illuminazione, il radar è ampiamente impiegato nel monitoraggio ambientale, nella geologia, nella cartografia e in molte altre applicazioni di telerilevamento.

I Persistent Scatterer, di cui si parlerà nei capitoli successivi, derivano dall'interferometria radar e vengono impiegati per il monitoraggio dei movimenti del suolo.

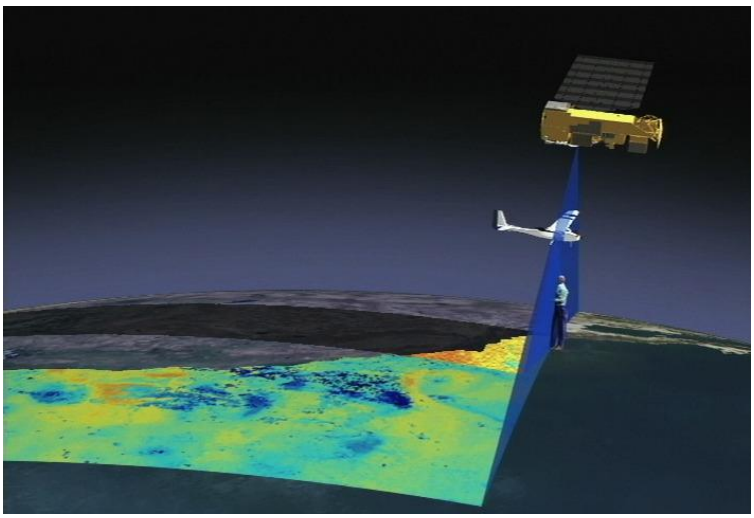


Figura 3: Remote sensing della superficie da satellite

Capitolo 1

1. Telerilevamento radar da Satellite

Il **remote sensing**, o telerilevamento, si distingue per l'utilizzo di un sistema di monitoraggio a distanza, composto da un sensore specifico installato su una particolare piattaforma. Le varie tipologie di telerilevamento si differenziano principalmente per le combinazioni uniche di sensori e piattaforme. In questo capitolo ci concentreremo sui sensori installati sui satelliti.

Indipendentemente dai sensori e dalle piattaforme impiegate, il remote sensing offre numerosi vantaggi per l’osservazione e il monitoraggio della Terra:

- acquisizioni sempre aggiornate;
- copertura di vaste aree;
- possibilità di ripetere le misurazioni;
- prospettiva privilegiata.

Tra le varie tecnologie di telerilevamento, un ruolo di primo piano è svolto dal Synthetic Aperture Radar (SAR), una forma avanzata di telerilevamento radar. Il SAR rappresenta una delle tecniche più sofisticate e versatili, distinguendosi per la sua capacità di superare alcuni dei limiti intrinseci ad altri sistemi di telerilevamento ottico e infrarosso.

1.1 Funzionamento del Synthetic Aperture Radar (SAR)

Il **SAR** è un tipo di raccolta di dati attiva in cui un sensore produce la propria energia e poi registra la quantità di quell'energia riflessa indietro dopo aver interagito con la superficie terrestre. Mentre l'imaging ottico è simile all'interpretazione di una fotografia, i dati SAR richiedono un modo diverso di pensare in quanto il segnale risponde invece alle caratteristiche superficiali come la struttura e l'umidità [5]. Questa interazione con la superficie terrestre è ottenuta grazie all'impiego di onde elettromagnetiche.

Il funzionamento alla base della tecnologia SAR è il seguente:

1. **Emissione di impulsi**: Il SAR emette una serie di brevi impulsi elettromagnetici, noti come microonde. Questi impulsi vengono inviati lateralmente, come è possibile osservare nella Figura 4 [7].

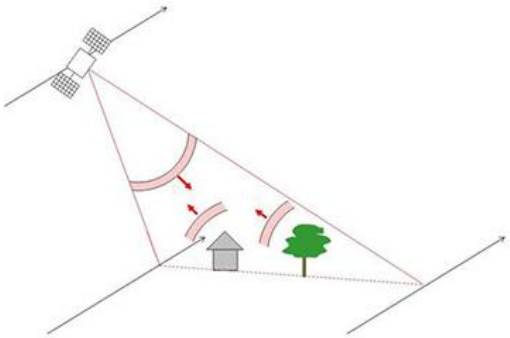


Figura 4: Emissione di impulso elettromagnetico da Satellite. Image Credit: ESA

2. **Riflessione e rilevamento**: Quando un impulso colpisce un oggetto sulla Terra, una parte di esso viene riflessa indietro verso l'antenna [14] . Il SAR registra il momento in cui l'impulso viene rinviato e la sua potenza, oltre alla fase delle microonde [7].

3. **Formazione dell'immagine**: Grazie a questo principio di osservazione laterale, il radar restituisce al sensore i segnali che colpiscono i diversi oggetti sulla Terra in momenti differenti [7] . Questo consente di distinguere gli oggetti colpiti dall'onda [7]

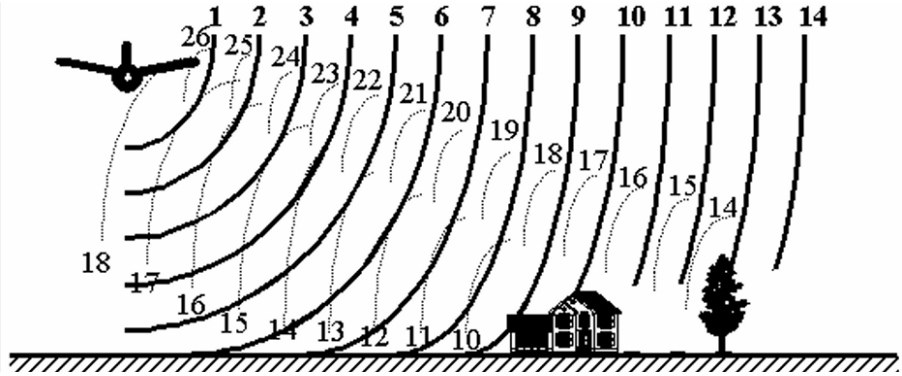


Figura 5: Concetto base dell'eco delle emissioni radar. Image Credit: Paul Messina, 1994

Ogni pixel nell'immagine radar rappresenta una quantità complessa dell'energia che è stata riflessa indietro verso il satellite [3]. La grandezza di ogni pixel, che rappresenta l'intensità dell'eco riflessa, può variare a seconda delle caratteristiche dell'oggetto che ha riflettuto l'onda, come la sua composizione, struttura, orientamento e rugosità della superficie.

4. **Interferometria radar**: Questi segnali di fase producono un interferogramma tra le due acquisizioni dei dati SAR. L'interferometria radar (InSAR) viene usata per misurare l'elevazione del terreno, mentre l'interferometria differenziale (DInSAR) viene usata per misurare gli spostamenti del suolo [7] .

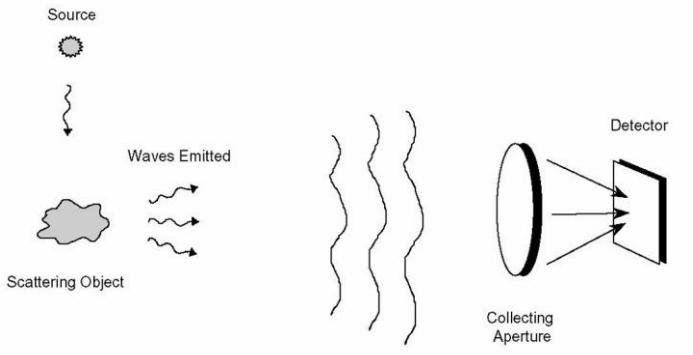


Figura 6: Sistema di Telerilevamento

1.2 Onde elettromagnetiche

La caratteristica distintiva di un sensore risiede nella sua abilità di catturare e rivelare informazioni trasmesse da ciò che può essere definito un messaggero. La fisica ci illustra che l'universo dispone di un'ampia varietà di messaggeri per comunicare: onde elettromagnetiche, particelle di materia, onde gravitazionali, eccetera. Nel campo del telerilevamento, come specificato precedentemente, il messaggero più comunemente impiegato è l'**onda elettromagnetica**.

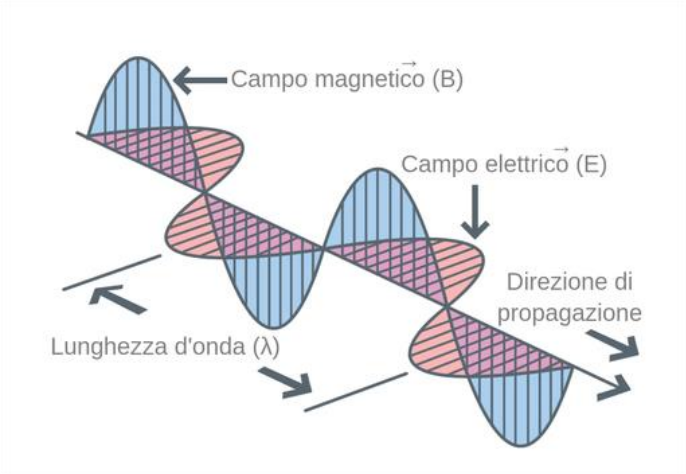


Figura 7: Propagazione dell'onda elettromagnetica

L'onda elettromagnetica è un'oscillazione sincronizzata del campo elettrico E e del campo magnetico B, tra loro ortogonali, che nel vuoto viaggia alla velocità della luce.

Queste onde possono essere suddivise in diverse regioni dello spettro elettromagnetico, ognuna con caratteristiche e applicazioni specifiche [17] . Lo spettro elettromagnetico rappresenta un intervallo continuo di lunghezze d'onda e frequenze della radiazione elettromagnetica. Come si vede nella Figura 8, a seconda della banda considerata, si utilizzano diversi nomi per le onde elettromagnetiche.

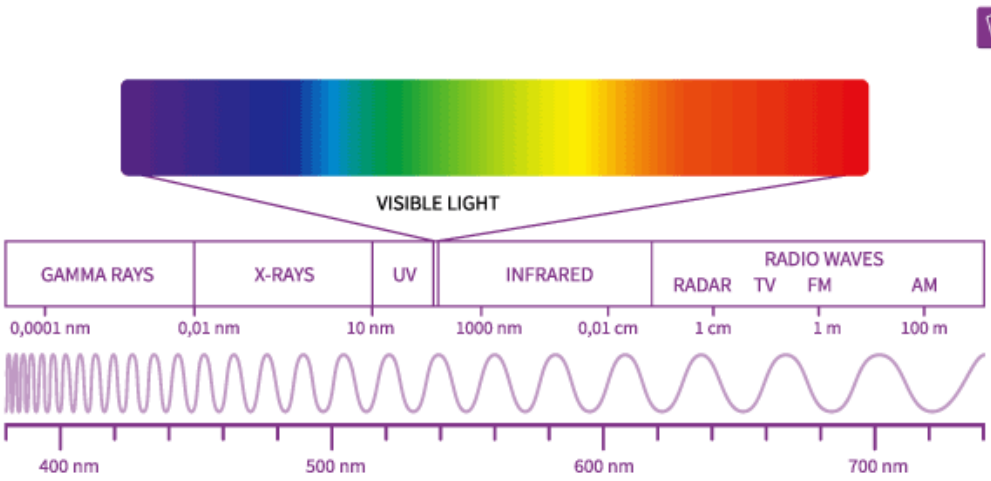


Figura 8: Spettro elettromagnetico

Lo spettro elettromagnetico inizia con le onde radio, caratterizzate dalle lunghezze d'onda più estese e dalle frequenze più basse. Man mano che ci si sposta lungo lo spettro, le lunghezze d'onda si accorciano e le frequenze aumentano. Si passa quindi attraverso diverse categorie di onde: le microonde, gli infrarossi, la luce visibile, i raggi ultravioletti (UV), i raggi X, fino a giungere ai raggi gamma, che presentano le lunghezze d'onda più corte e le frequenze più elevate.

Il diverso comportamento delle radiazione elettromagnetica, a diverse lunghezze d'onda, è dovuto al modo in cui la stessa radiazione interagisce con la materia.

È possibile avere:

- **riflessione speculare**: l'onda viene riflessa con lo stesso angolo di incidenza;
- **riflessione diffusa**: l'onda viene riflessa in tutte le direzioni;-
- **assorbimento**: l'onda viene assorbita dalla materia e trasformata in energia interna;
- **emissione**: di solito segue l'assorbimento. L'energia interna viene emessa sotto forma di onda elettromagnetica, ad una frequenza che pu` o differire da quella incidente;
- **trasmissione**: l'onda attraversa il mezzo cambiano direzione rispetto la normale alla superficie.

Tutti questi fenomeni coinvolgono sia la superficie terrestre che l'atmosfera (si veda la Figura 9) e risulta fondamentale conoscerli e studiarli per due ragioni: l'identificazione della natura del target e la scelta della banda operativa dei sensori montati a bordo dei satelliti.

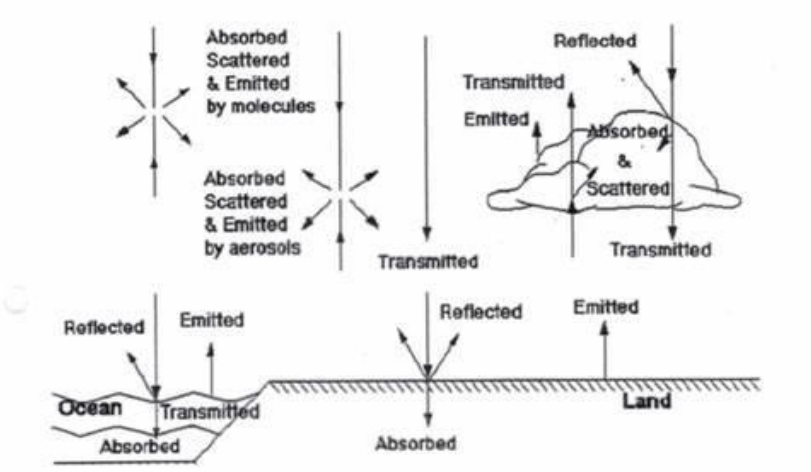


Figura 9: Interazione della radiazione con l'atmosfera

A causa dell'assorbimento della radiazione elettromagnetica, dovuto all'interazione radiazione-gas atmosferici, non tutte le regioni dello spettro sopracitate risultano utili al telerilevamento; solo in corrispondenza delle cosiddette finestre atmosferiche, o zone di trasparenza dell'atmosfera, la radiazione viene trasmessa e le perdite dovute all' assorbimento sono minime o nulle [18].

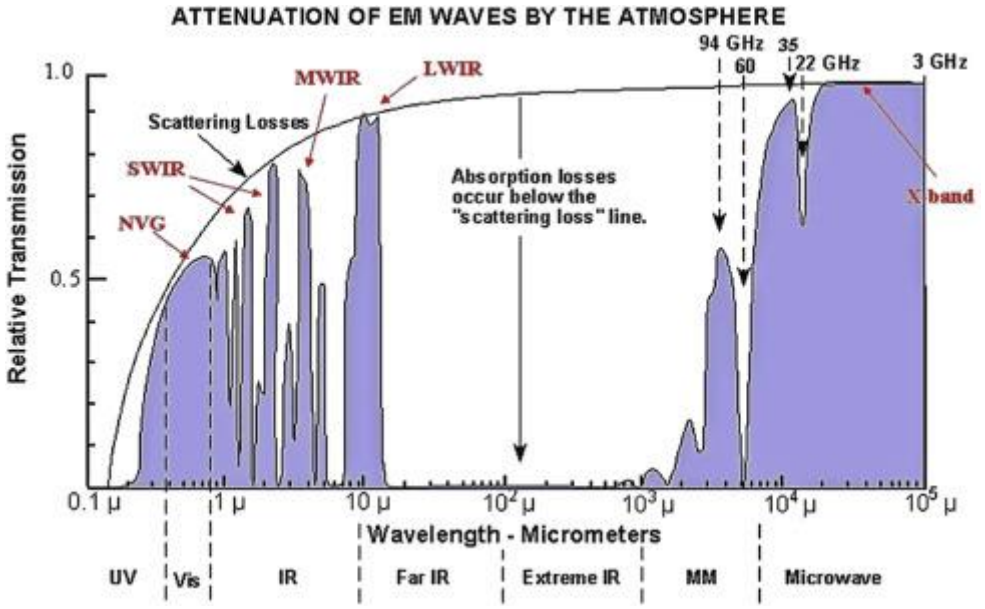


Figura 10: Attenuazione delle onde elettromagnetiche dall'atmosfera

La figura 10 evidenzia che la banda delle microonde ha la massima capacità di trasmissione, il che spiega perché i satelliti SAR (Synthetic Aperture Radar) utilizzano queste frequenze per le loro onde radar. Le frequenze della banda delle microonde variano da 1 a 300 GHz.

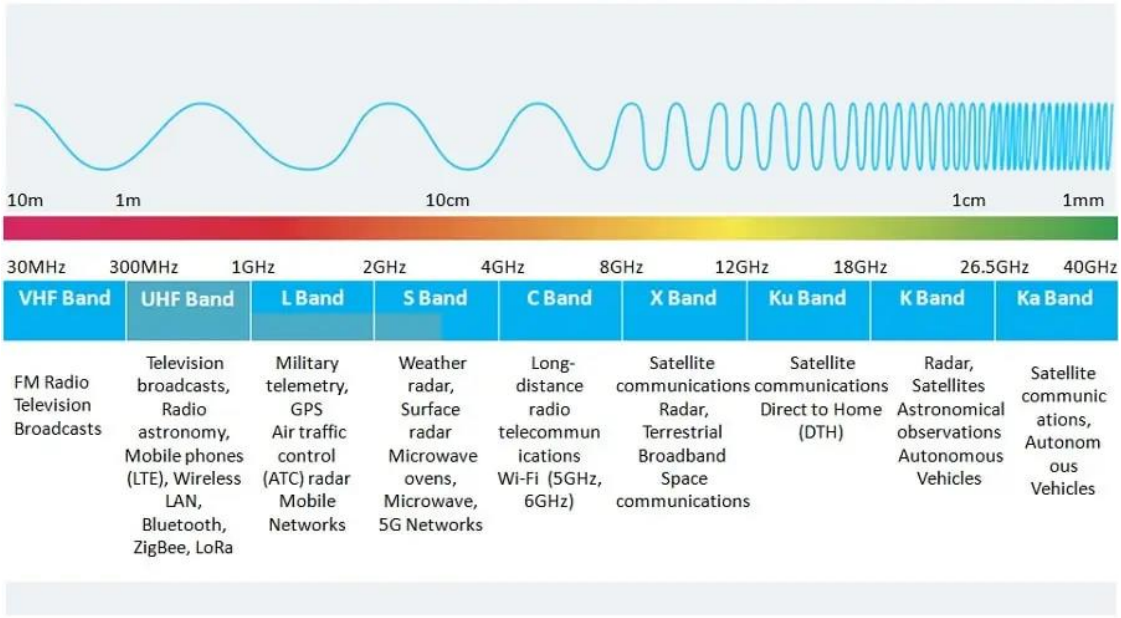


Figura 11: Bande delle microonde

Come possiamo osservare dalla Figura 11, le bande delle microonde possono essere suddivise in ulteriori bande, ognuna delle quali viene utilizzata per scopi precisi.

Le bande maggiormente utilizzate dai satelliti SAR per le immagini radar sono le seguenti:

- **Banda L (1-2 GHz):**
  La banda L è comunemente utilizzata per le missioni SAR a bassa risoluzione. Le onde radar in questa banda hanno una lunghezza d'onda più lunga, il che consente una migliore penetrazione attraverso le coperture vegetali e le nuvole. Questa banda è spesso utilizzata per monitorare cambiamenti nel suolo, come subsidenza o deformazioni [11].

- **Banda C (4-8 GHz):**
  La banda C è la più comune per i satelliti SAR. Le onde radar in questa banda offrono una buona risoluzione e una capacità di penetrazione adeguata attraverso l'atmosfera terrestre. È utilizzata per applicazioni come il monitoraggio delle risorse idriche, la mappatura delle foreste e la gestione del territorio [19].

- **Banda X (8-12 GHz):**
  Anche la banda X è utilizzata per missioni SAR ad alta risoluzione. Le onde radar in questa banda hanno una lunghezza d'onda più breve, consentendo una maggiore risoluzione. È ideale per il monitoraggio delle frane, la mappatura urbana e la rilevazione delle deformazioni del terreno [5] .

La lunghezza d'onda, come possiamo notare, è un'importante caratteristica da considerare quando si lavora con SAR, poiché determina come il segnale radar interagisce con la superficie e quanto lontano un segnale può penetrare in un mezzo. Ad esempio, un radar X-band, che opera a una lunghezza d'onda di circa 3 cm, ha scarso potere di penetrazione nella foresta a foglie larghe e interagisce principalmente con le foglie nella parte superiore della chioma degli alberi.

Un segnale L-band, d'altra parte, ha una lunghezza d'onda di circa 23 cm, ottenendo una maggiore penetrazione nella foresta e consentendo una maggiore interazione tra il segnale radar e rami grandi e tronchi degli alberi. La lunghezza d'onda non influisce solo sulla profondità di penetrazione nelle foreste, ma anche su altri tipi di copertura del suolo come il terreno e il ghiaccio [5].

1.3 Immagini SAR

Una immagine digitale SAR può essere vista come un mosaico (cioè un array bidimensionale formato da colonne e righe) di piccoli elementi di immagine (pixel). Ogni pixel è associato a una piccola area della superficie terrestre, ovvero la cella di risoluzione. Ogni pixel fornisce un numero complesso che contiene informazioni sull'ampiezza e sulla fase del campo a microonde retrodiffuso da tutti gli scatterer (rocce, vegetazione, edifici, ecc.) all'interno della corrispondente cella di risoluzione proiettata sul terreno. Diverse righe dell'immagine sono associate a diverse posizioni azimutali, mentre diverse colonne indicano diverse posizioni di slant range. La posizione e le dimensioni della cella di risoluzione nelle coordinate azimutali e di slant range dipendono solo dalle caratteristiche del sistema SAR [9].

L'immagine SAR rilevata contiene una misura dell'ampiezza della radiazione retrodiffusa verso il radar dagli oggetti (scatterer) contenuti in ciascuna cella di risoluzione SAR. Questa ampiezza dipende più dalla rugosità che dalla composizione chimica degli scatterer sul terreno.

Tipicamente, le rocce esposte e le aree urbane mostrano ampiezze elevate, mentre le superfici lisce e piatte (come le bacini d'acqua calma) mostrano ampiezze basse, poiché la radiazione viene principalmente riflessa lontano dal radar [9].

L'immagine SAR rilevata è generalmente visualizzata mediante livelli di scala di grigi, come mostrato nell'esempio della Figura 12. I pixel luminosi corrispondono ad aree di forte radiazione retrodiffusa (ad esempio, aree urbane), mentre i pixel scuri corrispondono a radiazioni retrodiffuse basse (ad esempio, un bacino d'acqua calma o vegetazione) [9].

Figura 12: Immagine SAR rilevata

Capitolo 2

2. Interferometria e Persistent Scatterers

Per misurare e monitorare le variazioni nella deformazione della superficie terrestre, esistono due tecniche generali: metodi geodetici e metodi geotecnici. L'approccio geodetico utilizza strumenti come la stazione totale, il livellamento preciso, il GPS e l'InSAR. Al contrario, la tecnica geotecnica utilizza strumenti come accelerometri, sismometri, laser, inclinometri, tiltmetro e micrometro [1]. In questo capitolo si discuterà sul funzionamento dell'approccio geodetico dell'InSAR e, più nello specifico, del PSInSAR.

2.1 InSAR: Interferometria Radar

L'Interferometria Radar (InSAR) rappresenta una estensione del SAR, introducendo una tecnica avanzata che sfrutta la coerenza di fase tra le immagini radar acquisite da diverse posizioni o in tempi diversi.

> La coerenza è la proprietà di un'onda elettromagnetica di mantenere una certa relazione di fase con se stessa durante la propagazione.

Più nello specifico, l' InSAR (Interferometria Radar a Sintesi d'Apertura), o Interferometria SAR, è una tecnica che sfrutta il cambiamento di fase del segnale tra due immagini acquisite nello stesso punto. Quando un punto sul terreno si muove, anche la distanza tra il sensore e il punto varia, influenzando di conseguenza il valore di fase registrato dal sensore.

La variazione di fase del segnale (Δφ) è espressa dall'equazione seguente:

$$\Delta\varphi = \frac{4\pi}{\lambda}\Delta R + \alpha$$

Dove λ è la lunghezza d'onda, ΔR è lo spostamento nella Line of Sight (LOS), e α è uno sfasamento di fase dovuto alle diverse condizioni atmosferiche al momento delle due acquisizioni radar [12].

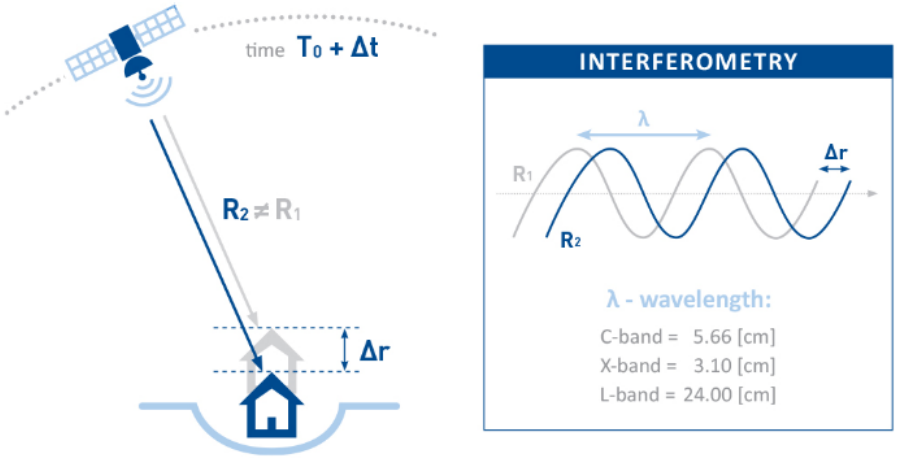


Figura 13: Funzionamenteo dell'InSAR.

La fase interferometrica (Δφ) è influenzata da quattro contributi:

1) **distorsioni topografiche** dovute agli angoli di visione leggermente diversi delle due passate del satellite (t)

2) **effetti atmosferici** (α) derivanti dalle distorsioni della lunghezza d'onda causate dallo strato umido,

3) eventuali **spostamenti** di intervallo del bersaglio radar (ΔR)

4) **rumore** (effetti di decorrelazione)

Questi fattori pertanto trasformano l'equazione precedentemente definita nella seguente:

$$\Delta\varphi = \frac{4\pi}{\lambda}\Delta R + \alpha + t + \text{noise}$$

2.2 PSInSAR: Persistent Scatterer Interferometric Synthetic Aperture Radar

La tecnica dell'Interferometria con Scatterer Persistenti (PSI) è uno dei primi approcci InSAR che coinvolge l'identificazione di pixel che hanno una riflessione forte e costante per un lungo periodo, chiamati Scatterer Persistenti (PS) o Scatterer Permanenti che sono identificati dalla loro alta coerenza o correlazione.

La tecnica PSInSAR utilizza un ampio set di interferogrammi che coprono la stessa area per identificare punti stabili al suolo che riflettono costantemente i segnali radar per un lungo periodo di tempo, tipicamente su diversi anni. Questi punti stabili si trovano solitamente su strutture artificiali come edifici, ponti, o su elementi naturali come rocce, scogliere e substrati rocciosi.

La tecnica PS si basa sull'identificazione di un gran numero di riflessioni radar provenienti dallo stesso punto e sull'uso di queste osservazioni per stimare la deformazione del suolo nel tempo.

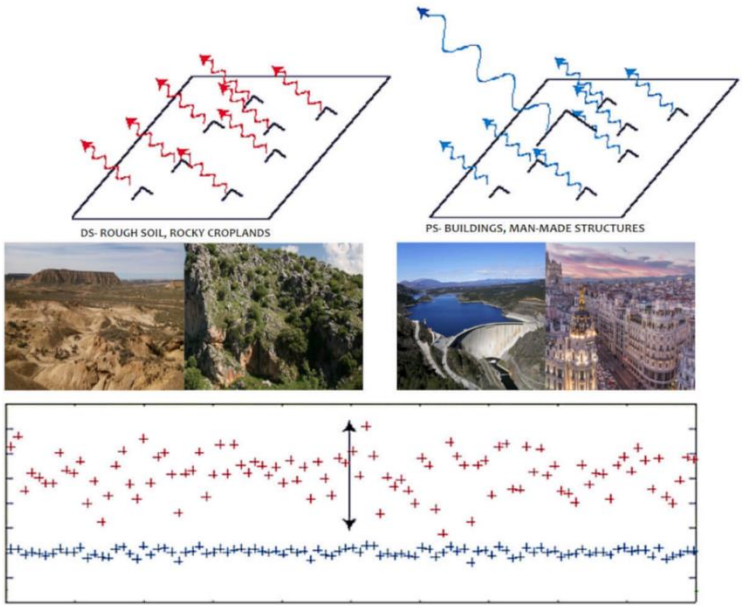


Figura 14: Strutture artificiali hanno una riflessione e coerenza maggiore rispetto aree non urbane

La tecnica PS-InSAR è molto efficace per misurare la deformazione della superficie in aree urbane e su altre strutture artificiali. Le aree caratterizzate da cambiamenti fisici, ad esempio aree vegetate, corpi d'acqua, dune mobili, zone di costruzione o distorsioni geometriche causate da pendii ripidi, avranno intrinsecamente meno PS. Inoltre, un PS altamente luminoso in un'area urbana può dominare i pixel adiacenti oltre il pixel corrispondente alla sua posizione reale.

Come è possibile anche osservare nella Figura 14, il principale svantaggio di questo metodo rappresenta la mancanza di densità di punti nelle aree non urbane. Non è adatto per aree che soffrono di de-correlazione temporale, ad esempio aree vegetate o aree in rapida urbanizzazione.

Per tale scopo è preferibile utilizzare la tecnica InSAR con Scatterer Distribuiti (DSInSAR) che utilizza una rete densa di scatterer, come la vegetazione, per stimare la deformazione del suolo.

Tale tecnica è utile per misurare la deformazione su vaste aree, come campi agricoli e regioni boschive, dove i riflettori stabili sono scarsi [12] .

2.2.1 Creazione della Serie Temporale del Displacement

L’obiettivo della tecnica PSInSAR è pertanto l’ottenimento di un ampio set di interferogrammi SAR per poter effettuare un monitoraggio della deformazione della superficie terrestre nel tempo.

Il workflow per la creazione delle serie temporali del displacement è il seguente:

2.2.1.1 Creazione della serie temporale

Come precedentemente delineato, i Diffusori Persistenti (PS) possono essere rilevati mediante l’osservazione dei pixel che manifestano un’intensità di riflessione robusta e stabile nel corso del tempo. Di conseguenza, una volta ottenuto l’insieme di interferogrammi SAR, questi PS vengono individuati sulla base della loro elevata coerenza.

2.2.1.2 Calcolo della Differenza di Fase

La discrepanza di fase tra le immagini radar raccolte in momenti distinti è modulata dalla fluttuazione della distanza tra il radar e il Diffusore Persistente (PS). Questa differenza di fase è strettamente connessa alla deformazione della superficie, dal momento che le variazioni nella distanza inducono mutamenti nella fase rilevata. In altre parole, ogni cambiamento nella distanza tra il radar e il PS si traduce in una variazione della fase misurata, permettendo così di monitorare le deformazioni della superficie terrestre.

2.2.1.3 Costruzione degli Interferogrammi

Gli interferogrammi sono generati sottraendo la fase delle immagini radar ottenute in tempi diversi. Ciascun interferogramma costituisce una rappresentazione della variazione di fase, che può essere trasformata in spostamenti della superficie lungo la linea di vista (LOS) del radar. In altre parole, ogni interferogramma fornisce una mappatura dettagliata delle differenze di fase, che possono essere interpretate come movimenti della superficie terrestre lungo la direzione di osservazione del radar.

2.2.1.4 Rimozione degli Effetti Atmosferici e Orbitali

Gli interferogrammi incorporano anche elementi che non sono correlati alla deformazione del suolo, come gli effetti atmosferici e le discrepanze orbitali. È necessario correggere questi effetti per isolare il segnale di deformazione. L'eliminazione di tali effetti è fondamentale per ottenere misurazioni precise della deformazione. In altre parole, per ottenere una stima accurata dello spostamento del terreno, è essenziale separare e rimuovere le componenti del segnale interferometrico che non sono legate alla deformazione del terreno.

2.2.1.5 Estrazione delle Serie Temporali

Una volta che si sono ottenuti gli interferogrammi corretti, la creazione delle serie temporali di spostamento avviene attraverso i seguenti passaggi:

Unwrapping della Fase: La fase radar misurata è confinata tra $-\pi$ e π. Per trasformare queste misure in spostamenti reali, è necessario il processo di unwrapping della fase, che elimina le ambiguità introducendo i corretti multipli di π .

Stima degli Spostamenti: Utilizzando un insieme di interferogrammi e considerando i Diffusori Persistenti (PS) come punti di riferimento, è possibile stimare gli spostamenti relativi. Questo viene fatto attraverso modelli matematici che calcolano il movimento della superficie nel tempo rispetto alla posizione iniziale.

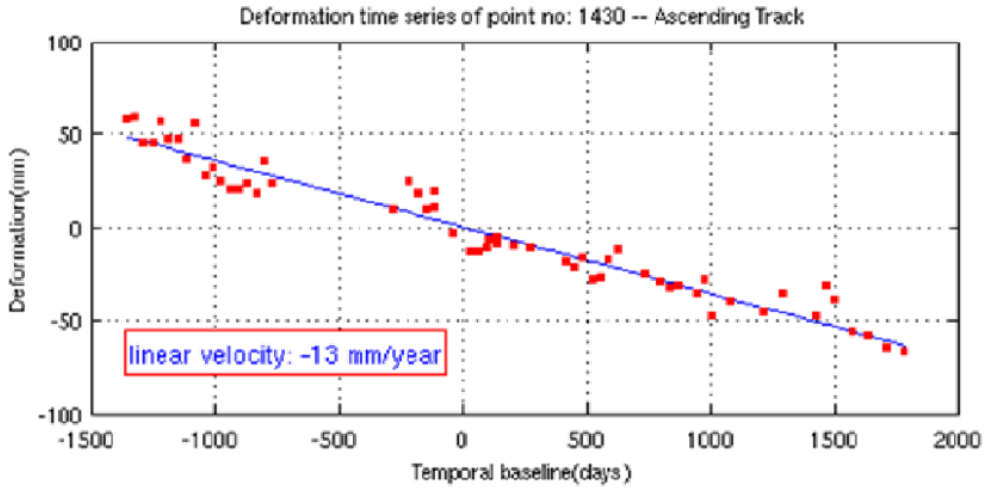


Figura 15: Esempio di una time series del displacement di un persistent scatterer. I punti rossi rappresentano le misurazioni nel tempo. La linea blu rappresenta il modello lineare che meglio si adatta alle diverse misurazioni.

2.2.2 Formazione dell'Interferogramma SAR

Per ottenere le informazioni interferometriche, le immagini devono essere distinte e poi combinate analiticamente. Una delle immagini, chiamata Master, rappresenta l'immagine di riferimento, solitamente la prima acquisita in una serie temporale; mentre la successiva (o le successive) che vengono acquisite in tempi diversi rispetto al master, è chiamata Slave.

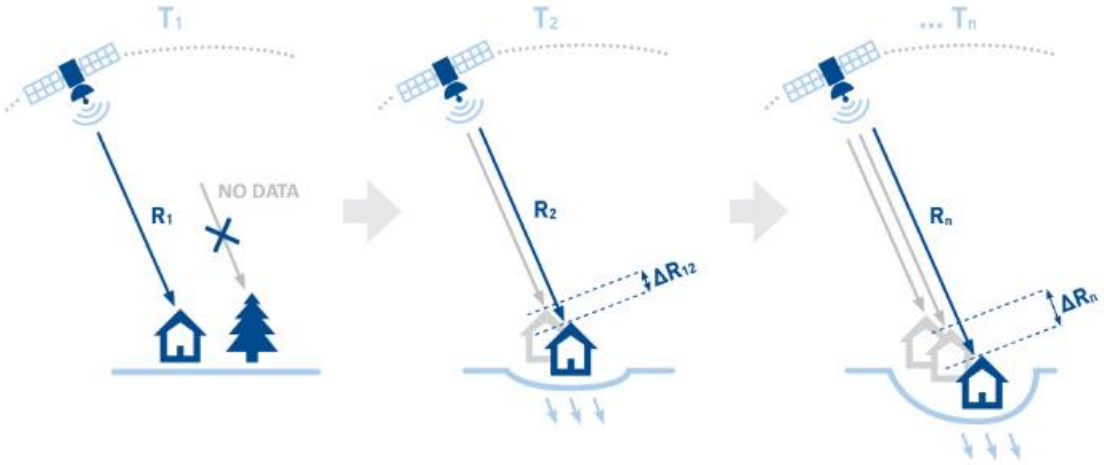


Figura 16: Processo di interferometria SAR per rilevare spostamenti della superficie terrestre nel tempo. Nella prima acquisizione al tempo T1 (Master) si può osservare come il satellite acquisisce l'immagine dell'edificio e non della vegetazione. La distanza dal satellite a un punto di riferimento sulla superficie è indicata come R1. Nella seconda acquisizione al tempo T2 (Slave) la distanza tra il satellite e lo stesso punto di riferimento è ora R2. Il cambiamento di distanza ΔR2 tra la prima e la seconda acquisizione è indicato. Questo cambiamento, come si può vedere, è dovuto a un movimento verticale della superficie. Lo stesso procedimento avviene con i successivi n Slave.

L'obiettivo dell'interferometria SAR è combinare queste immagini per rilevare differenze di fase dovute a cambiamenti nella scena osservata.

La fase di un pixel in un'immagine SAR complessa può essere espressa come:

$$\Phi = -\frac{4\pi}{\lambda} R + \Phi_{scatt}$$

dove:

- λ è la lunghezza d'onda del radar.
- R è la distanza tra il radar e il punto sulla superficie terrestre.
- Φ_{scatt} è la fase introdotta dalla superficie di diffusione (o scattering) nel punto considerato.

Per il **master**, la rappresentazione del pixel SAR complesso è:

$$u_m[i,k] = |u_m[i,k]| \cdot \exp(j\Phi[i,k])$$

Per lo **slave**, la rappresentazione è:

$$u_s[i,k] = |u_s[i,k]| \cdot \exp(j\Phi[i,k])$$

dove:

- $|u_s[i,k]|$ e $|u_m[i,k]|$ sono le ampiezze (moduli) dei pixel nelle immagini master e slave, rispettivamente.
- $\Phi_m[i,k]$ e $\Phi_s[i,k]$ sono le fasi dei pixel nelle immagini master e slave, rispettivamente.

L'interferogramma è ottenuto combinando le immagini master e slave come segue:

$$v[i,k] = u_m[i,k] \cdot u_s^*[i,k]$$

Dove $u_s^*[i,k]$ è il complesso coniugato del pixel slave. Questa operazione si può espandere come:

$$v[i,k] = |u_m[i,k]| \cdot |u_s[i,k]| \cdot \exp(j\Phi[i,k])$$

Qui, la fase risultante $\Phi[i,k]$ dell'interferogramma è la differenza di fase tra il master e lo slave:

$$\Phi[i,k] = \Phi_m[i,k] - \Phi_s[i,k]$$

2.3 Analisi della Time Series del Displacement dei PS

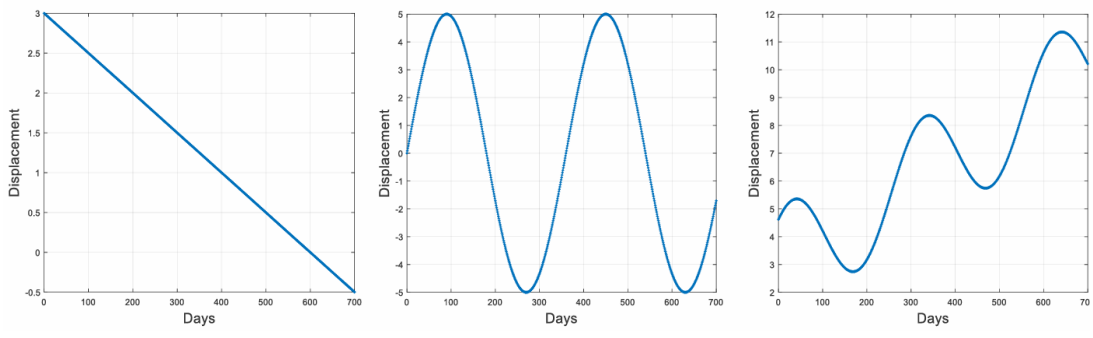
Una volta ottenuta la serie temporale dei displacement, è utile effettuare delle analisi per acquisire informazioni dettagliate sulla dinamica della deformazione del terreno.

Esaminando le serie temporali, si possono riconoscere modelli di deformazione, come spostamenti uniformi, accelerazioni, rallentamenti o fluttuazioni stagionali. Inoltre, queste serie sono frequentemente impiegate per individuare eventi particolari, poiché possono evidenziare fenomeni che provocano rapide deformazioni, come terremoti, frane o sprofondamenti.

2.3.1 Costruzione del Modello di Deformazione

La maggior parte degli algoritmi PSInSAR adottano un modello di regressione lineare per stimare le velocità medie. È possibile vedere un esempio di una serie temporale lineare osservando la Figura 16(a) .Tuttavia, poiché alcuni fenomeni di deformazione possono mostrare un comportamento più complesso nel tempo, l'uso di un modello di regressione lineare porta a stime potenzialmente errate della velocità media.

Per esempio, la velocità di una frana che si muove lungo un pendio ripido può essere influenzata dal contenuto idrico del materiale della frana, o una frana precedentemente inattiva può essere riattivata a seguito di un terremoto. In entrambi i casi, la serie temporale risultante non avrebbe un'inclinazione lineare costante, ma sarebbe piuttosto segmentata linearmente.



(a) *(b)* *(c)*

Figura 16: Esempi di grafici di serie temporali con tendenza lineare (a), componente stagionale (b) e tendenza lineare combinata con componente stagionale (c).

L'impiego di un modello di regressione lineare per stimare la velocità media su tutto l'arco temporale di osservazione potrebbe non rivelare informazioni cruciali, come ad esempio un rallentamento nel processo di deformazione. Le faglie possono alternare stati di attività e dormienza, il che implica che la velocità media di un'area specifica potrebbe essere sottostimata o sovrastimata. Inoltre, un cambiamento periodico tra subsidenza e sollevamento, dovuto alla variazione del contenuto idrico del suolo, genera un segnale periodico. Un esempio di tale serie temporale periodica è illustrato nella Figura 16(b). In questo scenario, la stima della pendenza della regressione lineare, ovvero la velocità media, risulterebbe essere zero. Di conseguenza, un algoritmo che identifica le aree in deformazione attiva basandosi sulla velocità media non riconoscerebbe quest'area come attiva.

Inoltre, è possibile che diversi modelli di deformazione si sovrappongano, come evidenziato nella Figura 16(c). La serie temporale mostra una tendenza lineare e una componente stagionale. Un esempio di tale deformazione è rappresentato dalla deformazione di una diga di recente costruzione. Il corpo della diga subisce una subsidenza a causa dell'assestamento dei materiali da costruzione e del peso dell'acqua del serbatoio. Il livello dell'acqua del serbatoio può variare nel tempo a causa delle fluttuazioni nell'afflusso d'acqua e del consumo d'acqua per l'irrigazione, l'uso di acqua potabile o la produzione di energia. Questa variazione del livello dell'acqua può portare a un cambiamento periodico tra subsidenza e sollevamento.

La realizzazione di un metodo per la decomposizione della serie temporale, al fine di eliminare la componente stagionale e identificare il trend, sarà oggetto di discussione nei capitoli successivi. Questo approccio permetterà di isolare e analizzare più efficacemente i modelli di deformazione del terreno.

Capitolo 3

3. Architettura del sistema

Questo capitolo si concentra sull'architettura di un sistema proposto, progettato per orchestrare microservizi tramite Docker Swarm. L'obiettivo è pertanto monitorare il displacement dei Persistent Scatterer, riducendo le fluttuazioni periodiche causate da fattori stagionali.

Inizieremo fornendo una panoramica dettagliata dei principali componenti del sistema e delle loro interazioni. Successivamente, esploreremo le decisioni progettuali che hanno portato all'adozione di Docker Swarm come framework di orchestrazione, evidenziando i vantaggi derivanti dall'utilizzo di microservizi per la gestione modulare delle funzionalità di monitoraggio e filtraggio dei dati.

Questa architettura non solo facilita la scalabilità e la gestione dei microservizi, ma promuove anche un approccio robusto e flessibile nel trattare con le complesse dinamiche dei dati geofisici.

3.1 Docker

Docker è una piattaforma open source dedicata allo sviluppo, alla distribuzione e all'esecuzione di applicazioni. Permette di isolare le applicazioni dall'infrastruttura sottostante, facilitando così la distribuzione rapida del software. Con Docker, è possibile gestire l'infrastruttura in modo simile alla gestione delle applicazioni stesse. Sfruttando le metodologie di Docker per il deployment, il testing e la distribuzione del codice, è possibile ridurre in modo significativo il gap temporale tra lo sviluppo del codice e la sua esecuzione in ambiente di produzione.

3.1.1 Concetti base

Docker offre la possibilità di creare e eseguire applicazioni all'interno di un ambiente chiamato **container**, basato su **immagini** Docker. Le immagini Docker sono il risultato di una definizione specifica contenuta in un file chiamato **Dockerfile**. Questo file contiene le istruzioni necessarie per creare un'immagine Docker, inclusi il sistema operativo base, le dipendenze software e il codice dell'applicazione.

Questo approccio permette un isolamento leggero e sicuro, consentendo di eseguire numerosi container simultaneamente su un singolo host, indipendentemente dalla configurazione specifica dell'host stesso. Le immagini Docker sono facilmente riproducibili e possono essere condivise attraverso registri di immagini come Docker Hub.

Utilizzando Docker e il Dockerfile, gli sviluppatori possono definire in modo preciso e riproducibile l'ambiente di esecuzione delle loro applicazioni. Questo facilita la condivisione tra i team di sviluppo e garantisce che tutti i collaboratori utilizzino la stessa versione funzionante dell'applicazione, riducendo così le differenze tra ambienti di sviluppo, test e produzione.

Docker fornisce strumenti e una piattaforma per gestire il ciclo di vita dei contenitori:

- Sviluppo dell'applicazione e dei suoi componenti supportati tramite container.
- Utilizzo del container come unità per la distribuzione e il testing dell'applicazione.
- Distribuzione dell'applicazione in un ambiente di produzione, che può essere un data center locale, un provider cloud o entrambi, utilizzando contenitori o servizi orchestrati.

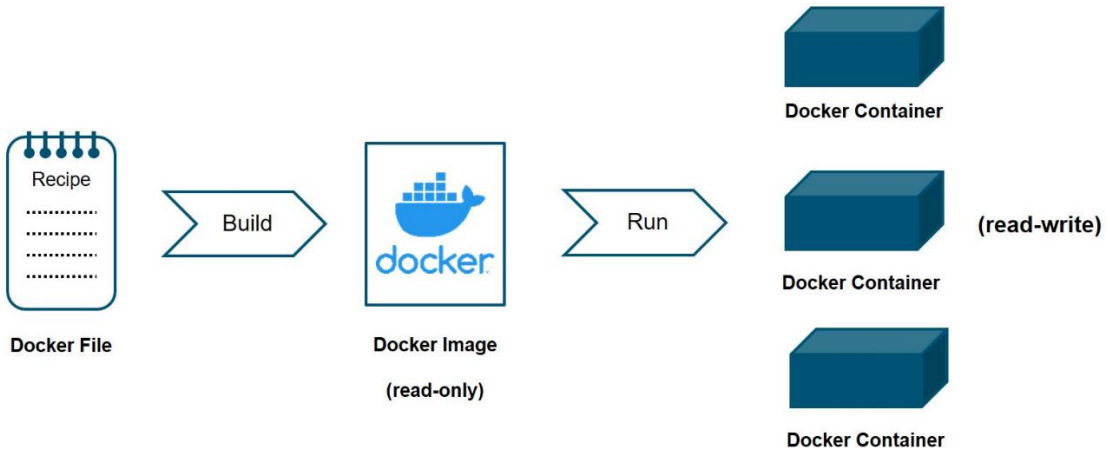


Figura 17: Containerizzazione delle immagini Docker

3.1.2 Architettura di Docker

Docker utilizza un'architettura client-server. Il client Docker interagisce con il Docker daemon, il quale gestisce la costruzione, l'esecuzione e la distribuzione dei contenitori Docker. Il client Docker e il daemon possono essere eseguiti sullo stesso sistema oppure è possibile connettere un client Docker a un Docker daemon remoto. La comunicazione tra il client Docker e il daemon avviene tramite un'API REST, che può utilizzare un UNIX socket o un'interfaccia di rete.

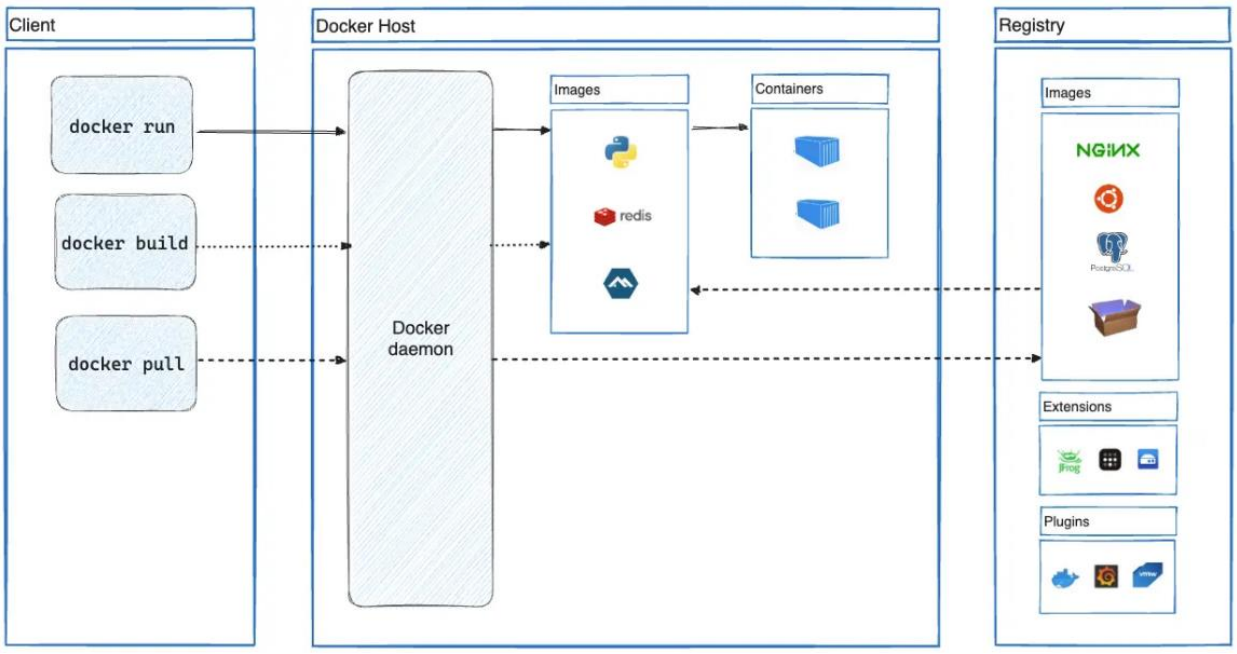


Figura 18: Architettura di Docker

3.1.3 Docker Swarm: deployment e orchestrazione

La containerizzazione consente di trasferire e scalare applicazioni su cloud e data center. I container garantiscono che le applicazioni funzionino sempre allo stesso modo, indipendentemente dall'ambiente in cui vengono eseguite, facilitando l'utilizzo rapido e agevole di diverse infrastrutture. Man mano che le applicazioni crescono, è necessario disporre di strumenti che automatizzino la manutenzione, sostituiscano automaticamente i container non funzionanti e gestiscano il rilascio di aggiornamenti e riconfigurazioni dei container durante il loro ciclo di vita. Gli strumenti che gestiscono, scalano e mantengono applicazioni containerizzate sono chiamati **orchestratori**.

Un orchestratore fornito da Docker è **Docker Swarm**, il quale consente di gestire un cluster di nodi come un singolo sistema logico. Docker Swarm facilita il deployment automatizzato, il bilanciamento del carico e la gestione della scalabilità dei container, garantendo che le applicazioni rimangano affidabili e efficienti nel loro funzionamento. Swarm si occupa di coordinare i container, ridistribuendo le risorse in caso di fallimenti e gestendo le operazioni di aggiornamento e riconfigurazione in modo controllato e senza interruzioni.

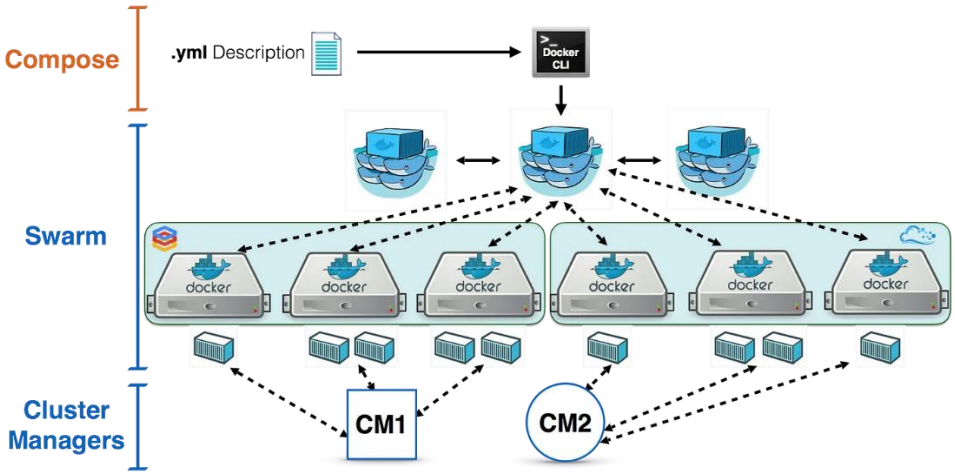


Figura 19: Esempio di Docker Swarm

Tutte le attività in Swarm vengono programmate come **servizi**, che rappresentano gruppi scalabili di container dotati di funzionalità di rete aggiuntive, gestiti automaticamente da Swarm.

Unoltre, tutti gli oggetti Swarm devono essere descritti in manifesti noti come stack file. Questi file YAML specificano tutti i componenti e le configurazioni della propria applicazione Swarm, e consentono di creare e distruggere l'applicazione in qualsiasi ambiente Swarm.

3.1.4 Nodi in Docker Swarm: Differenze tra Manager e Worker

In Docker Swarm, i **nodi** sono le unità fondamentali che compongono il cluster, formando insieme un sistema distribuito per eseguire e gestire i container. Ogni nodo può essere configurato come **manager** o **worker**, e ciascuno di questi ruoli ha responsabilità specifiche.

Un nodo in Docker Swarm è un host fisico o virtuale che partecipa al **cluster**. Questo host esegue Docker e fornisce risorse come CPU, memoria e storage per eseguire i container. I nodi sono gli elementi che consentono di scalare le applicazioni su più macchine, garantendo che i container possano essere eseguiti in modo distribuito.

I **manager** sono i nodi che si occupano di coordinare e gestire l'intero cluster Docker Swarm.

I nodi manager si occupano delle seguenti operazioni:

- **Orchestrazione**: Decidono dove e come eseguire i container. Se un container si arresta inaspettatamente, i manager assegnano un nuovo container a uno dei nodi disponibili.
- **Stato del Cluster**: Tengono traccia dello stato del cluster, monitorando le prestazioni e rilevando eventuali problemi.
- **Aggiornamenti**: Gestiscono gli aggiornamenti dei servizi senza interruzioni, garantendo che le nuove versioni dei container siano distribuite in modo fluido.
- **API Swarm**: Forniscono l'interfaccia attraverso cui si interagisce con il cluster, come l'invio di comandi per creare nuovi servizi o aggiornare quelli esistenti.

I **worker** sono i nodi che eseguono effettivamente i container come indicato dai manager.

Si occupano pertanto di:

- **Esecuzione dei Container**: Ricevono i task dai manager e ospitano i container che compongono i vari servizi dell'applicazione.
- **Segnalazione dello Stato**: Comunicano il loro stato ai manager, fornendo informazioni utili per l'orchestrazione e la gestione del cluster.

I worker eseguono i compiti assegnati senza partecipare alle decisioni sullo stato del cluster. Essi eseguono semplicemente i container e riportano l'avanzamento delle attività.

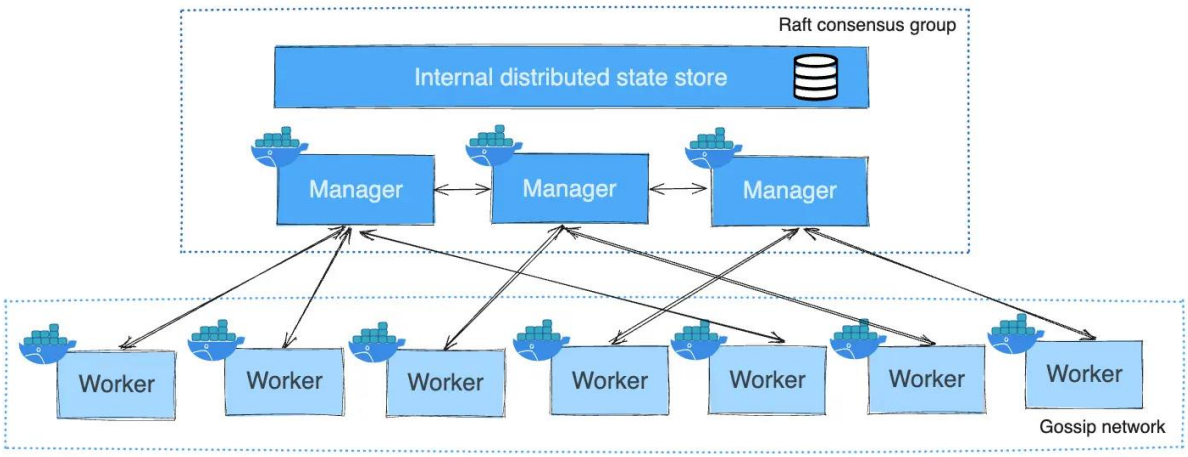


Figura 20: Swarm diagram

3.1.5 Docker Networks

Una volta definito Docker Swarm come orchestratore per la gestione automatizzata dei container, diventa fondamentale considerare la **comunicazione** tra i servizi. In un ambiente Swarm, i servizi distribuiti su diversi nodi devono interagire tra loro per garantire il corretto funzionamento dell'applicazione complessiva. Per facilitare questa comunicazione, Docker utilizza una funzionalità chiamata **Docker Network**.

Docker Network fornisce un'infrastruttura di rete virtuale che consente ai container e ai servizi in Docker Swarm di comunicare tra loro. La rete Docker crea un ambiente connesso in cui i container possono scambiarsi dati, inviare richieste e rispondere, indipendentemente dalla loro posizione fisica sui nodi del cluster. Questo è essenziale per costruire applicazioni distribuite complesse dove i diversi componenti devono collaborare strettamente.

In un cluster Swarm, la comunicazione tra i servizi è gestita principalmente tramite **overlay network**, che permette ai container di diversi servizi, dislocati su vari nodi, di comunicare come se fossero sulla stessa rete locale. Questa rete overlay garantisce che i container possano trovare e comunicare con altri servizi senza configurazioni di rete complesse, utilizzando nomi DNS di servizio per risolvere i container.

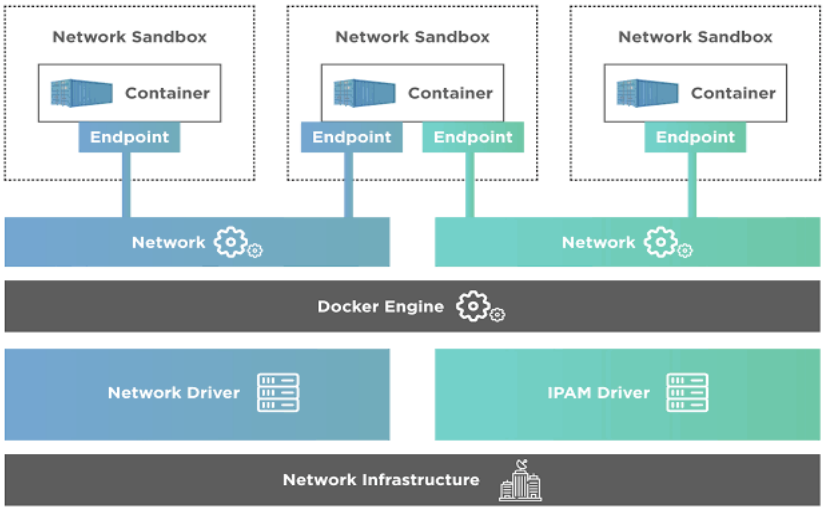


Figura 20: Architettura del Container Networking Model. Si può osservare come i Container 1 e 2 sono in grado di comunicare tra loro in quanto nella stessa rete. Medesimo caso per i Container 2 e 3. I Container 1 e 3 non saranno in grado in comunicare in quanto si trovano in reti differenti.

3.2 Componenti del Sistema

L'architettura proposta per il monitoraggio e l'analisi stagionale dei displacement dei Persistent Scatterers si basa su una serie di microservizi, ciascuno con una funzione specifica all'interno del sistema. Questi microservizi, orchestrati da Docker Swarm, cooperano per gestire la raccolta, l'elaborazione, la memorizzazione e la visualizzazione dei dati, garantendo un flusso continuo e affidabile di informazioni. In questa sezione, verranno descritti in dettaglio i componenti principali del sistema, fornendo una comprensione chiara dei loro ruoli e delle loro interazioni.

I servizi sono organizzati in tre stack principali per una gestione più efficiente e modulare.

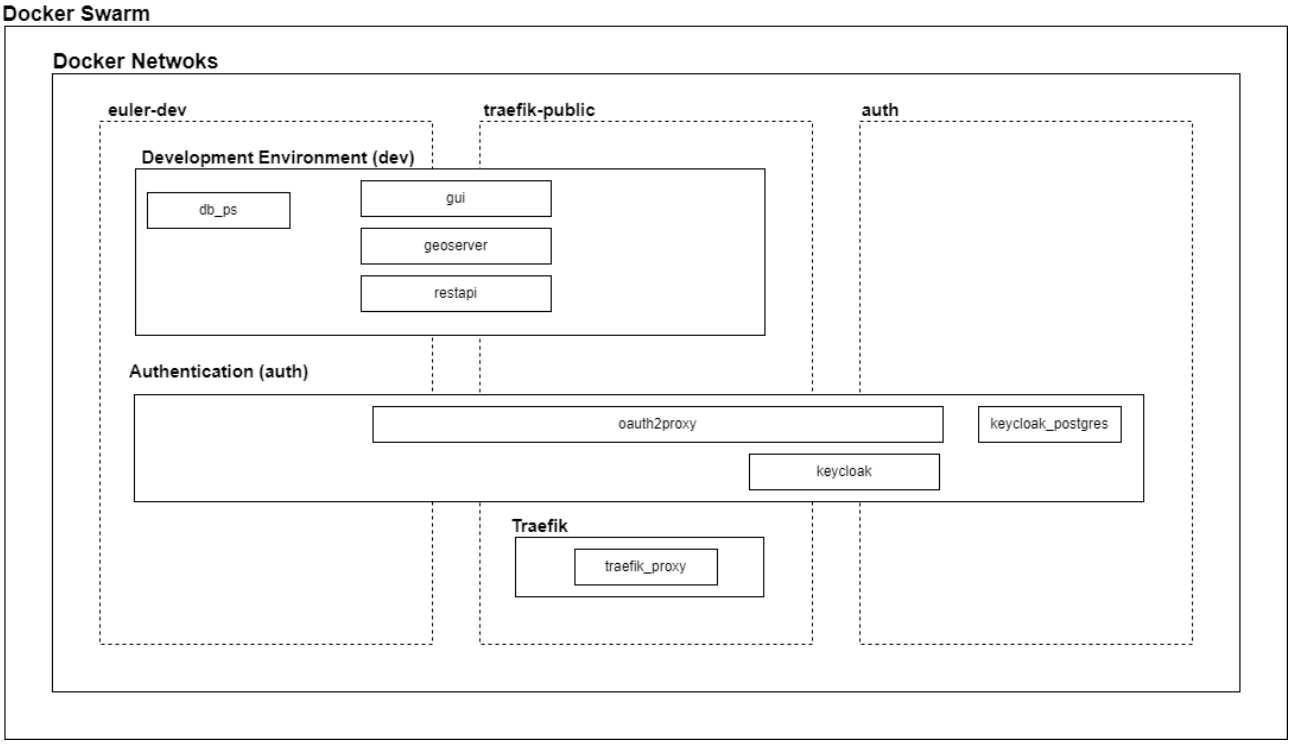


Figura 21: Architettura del Sistema.

3.2.1 Development Environment (stack-dev)

Questo stack comprende i componenti chiave per lo sviluppo e l'operatività dell'ambiente di monitoraggio dei Persistent Scatterers. Include i servizi che gestiscono l'interfaccia utente, la memorizzazione e l'elaborazione dei dati geospaziali.

3.2.1.1 GUI

La GUI (Graphical User Interface) rappresenta l'interfaccia grafica con cui gli utenti interagiscono direttamente. È progettata per fornire un’esperienza utente intuitiva, consentendo la visualizzazione e l'analisi dei dati geospaziali relativi ai displacement dei PS. La GUI permette agli utenti di visualizzare i Persistent Scatterers sulla mappa e di esplorare i dati spaziali e interpretare i dati sui displacement direttamente sulla mappa interattiva. Inoltre integra un modulo per l’analisi delle serie temporali che permette agli utenti di filtrare le fluttuazioni stagionali dai dati di displacement. Questo modulo consente di visualizzare trend più accurati, migliorando l'interpretazione dei dati.

3.2.1.2 Geoserver

GeoServer è un software gratuito e open source basato su Java che consente agli utenti di visualizzare e modificare dati geospaziali. Utilizzando gli standard aperti stabiliti dal Consorzio Geospaziale Aperto (OGC), GeoServer offre una grande flessibilità nella creazione di mappe e nella condivisione dei dati.

GeoServer permette di mostrare le informazioni spaziali al mondo intero. Implementando lo standard Web Map Service (WMS), GeoServer può generare mappe in vari formati di output. OpenLayers, una libreria di mappatura gratuita, è integrata in GeoServer, facilitando e velocizzando la creazione di mappe. GeoServer è costruito su GeoTools, un toolkit GIS open source in Java.

Conformandosi agli standard Web Feature Service (WFS) e Web Coverage Service (WCS), permette la condivisione e la modifica dei dati utilizzati per generare le mappe. Inoltre, utilizza lo standard Web Map Tile Service (WMTS) per suddividere le mappe pubblicate in riquadri, facilitandone l'uso in applicazioni web e mobili [2] .

GeoServer supporta la maggior parte delle proiezioni del database EPSG e può riproiettare su qualsiasi di queste su richiesta, consentendo a client con limitate capacità di riproiezione di passare il carico sul server [10] .

Nel contesto del nostro sistema, Geoserver è stato utilizzato per la pubblicazione e la gestione dei dati geospaziali. Ha fornito servizi di mappa e funzioni WMS, permettendo alla GUI di ottenere dati spaziali per la visualizzazione e analisi dei PS.

3.2.1.3 PostGIS

PostGIS è una estensione spaziale per il database PostgreSQL. PostgreSQL è un potente sistema di database relazionale a oggetti con oltre 35 anni di sviluppo attivo. Tale DBMS open source ha guadagnato una solida reputazione per la sua affidabilità, robustezza delle funzionalità e prestazioni. PostGIS aggiunge a PostgreSQL funzionalità geospaziali, consentendo di archiviare, indicizzare e interrogare dati geospaziali [15] .

Nel contento del nostro sistema, PostGIS è stato utilizzato per memorizzare e gestire i dati geospaziali relativi ai Persistent Scatterers.

3.2.1.4 REST API

Un’API REST (chiamata anche API RESTful o API Web RESTful) è un’application programming interface (API) conforme ai principi di progettazione di stile architetturale REST, o representational state transfer. I principi di progettazione REST che sono stati utilizzati nella definizione del nostro modulo sono i seguenti:

- **Interfaccia uniforme**: Tutte le richieste API per la stessa risorsa devono avere lo stesso aspetto, indipendentemente dalla provenienza della richiesta. L’API REST garantisce che lo stesso dato, come il nome o l’indirizzo e-mail di un utente, appartenga a un solo URI (Uniform Resource Identifier).

- **Disaccoppiamento client-server**: Le applicazioni client e server devono essere completamente indipendenti l’una dall’altra. L’applicazione client conosce solo l’URI della risorsa richiesta e non può interagire con l’applicazione server in altri modi.

- **Condizione di stateless**: Le API REST sono stateless, il che significa che ogni richiesta deve includere tutte le informazioni necessarie per elaborarla. Non richiedono sessioni lato server.

L'implementazione di una API REST ha apportato significativi vantaggi, tra cui una maggiore flessibilità e leggerezza nell'integrazione tra applicazioni [4] . Inoltre, ha facilitato la connessione tra i vari componenti della nostra architettura a microservizi.

Nel contesto del nostro sistema, l’API REST ha gestito le diverse richieste verso GeoServer.

3.2.2 Authentication (stack-auth)

Questo stack gestisce i componenti responsabili dell'autenticazione e dell'autorizzazione, assicurando che l'accesso ai diversi servizi sia sicuro e controllato.

3.2.2.1 OAuth2 Proxy

OAuth2Proxy è uno strumento open source utilizzato per fornire autenticazione e autorizzazione mediante provider **OAuth 2.0** per applicazioni web e API. Funziona come un reverse proxy che integra autenticazione basata su OAuth, facilitando l'accesso sicuro a servizi senza la necessità di implementare complessi meccanismi di autenticazione all'interno delle applicazioni stesse. OAuth 2.0, che significa "Open Authorization" (Autorizzazione Aperta), è uno standard progettato per consentire a un sito web o a un'applicazione di accedere alle risorse ospitate da altre app web per conto di un utente. Questo sistema fornisce un accesso regolamentato e specifica quali azioni l'applicazione client può eseguire su tali risorse, evitando di condividere le credenziali dell'utente.

OAuth 2.0 è un metodo per l'autorizzazione, non per l'autenticazione. Il permesso di accedere alle risolre per conto dell'utente finale è ottenuto mediante i Token di Accesso che vengono utilizzati da OAuth 2.0 [20] .

OAuth 2.0 è un framework di autorizzazione che si basa su un sistema di ruoli distinti. Questi ruoli sono fondamentali per comprendere il funzionamento di OAuth 2.0 e per garantire una gestione sicura e organizzata delle autorizzazioni e degli accessi. I ruoli principali in un sistema OAuth 2.0 sono i seguenti:

- **Proprietario della Risorsa (Resource Owner)** : è l'entità (tipicamente un utente o un sistema) che possiede le risorse protette. Queste risorse possono essere dati personali, file, o qualsiasi altro tipo di informazione sensibile. Il Proprietario della Risorsa ha il potere di concedere o revocare l'accesso a queste risorse. In altre parole, è l'entità che ha il controllo sui permessi di accesso ai dati o alle funzionalità protette.

- **Client**: è l'entità che desidera accedere alle risorse protette. Per ottenere l'accesso, il Client deve ottenere un Token di Accesso appropriato dall'Authorization Server. Il Client può essere un'applicazione web, un'app mobile, o un qualsiasi software che richiede accesso alle risorse del Proprietario della Risorsa.

- **Server di Autorizzazione (Authorization Server)**: è il componente che gestisce il processo di emissione dei Token di Accesso al Client. Questo server autentica il Proprietario della Risorsa e ottiene il consenso per rilasciare i Token di Accesso.

- **Server delle Risorse (Resource Server)**: è il componente che ospita le risorse protette e risponde alle richieste di accesso provenienti dal Client. Questo server convalida i Token di Accesso forniti dal Client e, se validi, consente l'accesso alle risorse richieste.

Nel contesto del nostro sistema Oauth2proxy è stato utilizzato come intermediario tra gli utenti e la GUI, facilitando l'integrazione con OAuth2 per la gestione sicura dell'accesso e delle autorizzazioni.

3.2.2.2 Keycloak

Keycloak è un software per la gestione delle identità e degli accessi (IAM), che si occupa della gestione di chi può accedere alle tue applicazioni e ai tuoi sistemi e in che modo. Ecco una panoramica delle sue principali funzionalità:

- **Autenticazione**: Keycloak si occupa dell'autenticazione degli utenti, quindi non è necessario integrare sistemi di login specifici per ciascuna applicazione. Quando un utente accede a un servizio, viene reindirizzato a Keycloak per l'autenticazione. Una volta autenticato, l'utente può accedere ai servizi senza dover effettuare il login nuovamente per ogni applicazione.

- **Single Sign-On (SSO)**: Con il Single Sign-On (SSO), una volta che gli utenti hanno effettuato il login tramite Keycloak, possono accedere a tutte le applicazioni collegate senza dover inserire le loro credenziali ogni volta. Questo vale anche per il logout: una volta che escono da una delle applicazioni, vengono automaticamente scollegati da tutte le altre.

- **Supporto ai Protocolli Standard**: Keycloak è compatibile con protocolli standard per l'autenticazione e l'autorizzazione come OpenID Connect, OAuth 2.0, e SAML, garantendo la possibilità di integrarsi facilmente con altre applicazioni e servizi che supportano questi protocolli.

- **Autorizzazione Granulare**: Oltre alla gestione delle autorizzazioni basata sui ruoli, Keycloak offre funzionalità avanzate per la definizione di politiche di accesso dettagliate. Questo permette di controllare con precisione quali utenti possono accedere a quali risorse e quali operazioni possono eseguire [13].

Fornisce la gestione delle identità e l'autenticazione, consentendo il controllo sicuro dell'accesso ai vari microservizi. Gestisce l'autenticazione degli utenti e le politiche di accesso.

3.2.2.3 Keycloak Postgres

Il backend di database per Keycloak, che memorizza le informazioni sugli utenti, le sessioni e le politiche di accesso. Assicura la persistenza dei dati di autenticazione.

3.2.3 Traefik (stack-traefik)

Questo stack gestisce traefik proxy, responsabile per l’instradamento e load balancing del sistema.

3.2.3.1 Traefik proxy

Traefik è un reverse proxy e load balancer progettato per facilitare la gestione dinamica del traffico di rete in ambienti cloud-native e containerizzati.

È stato sviluppato con un focus particolare sulla facilità d'uso, l'integrazione nativa con piattaforme di orchestrazione come Docker, Kubernetes, e Swarm, e il supporto per il routing dinamico basato su configurazioni dinamiche e aggiornamenti automatici.

Le caratteristiche principali di Traefik sono le seguenti:

- **Routing dinamico**: Traefik è in grado di rilevare automaticamente nuovi servizi all'interno dell'infrastruttura, grazie alla sua integrazione con le API dei provider cloud e delle piattaforme di orchestrazione. Questo significa che è in grado di aggiornare dinamicamente le sue configurazioni per includere nuovi servizi senza la necessità di riavvii o interventi manuali.

- **Load balancing**: Supporta il bilanciamento del carico tra diverse istanze di servizio, distribuendo il traffico in base a regole configurabili (come round-robin, session persistence, o weighted).

- **Gestione automatica dei certificati SSL/TLS**: Traefik può automatizzare il processo di gestione dei certificati SSL/TLS tramite l'integrazione con Let's Encrypt. Questo significa che può generare, rinnovare e applicare automaticamente certificati SSL/TLS per i domini gestiti senza bisogno di intervento manuale.

- **Integrazione con container e orchestratori**: È progettato per funzionare in ambienti containerizzati come Docker e orchestration platforms come Kubernetes e Swarm, dove può agire come un ingress controller per gestire il traffico HTTP e HTTPS verso i container.

- **Middleware e regole di routing avanzate**: Traefik supporta l'uso di middleware per applicare regole di routing avanzate, come la ridirezione URL, l'autenticazione basata su token, la limitazione della velocità di richiesta, e altro ancora. Questi middleware possono essere configurati e applicati dinamicamente alle richieste in base alle esigenze specifiche dell'applicazione.

- **Interfaccia utente e dashboard**: Fornisce un'interfaccia utente web per la configurazione e il monitoraggio delle regole di routing, dei backend e delle statistiche di traffico. Questa dashboard semplifica la gestione e il monitoraggio delle configurazioni di Traefik.

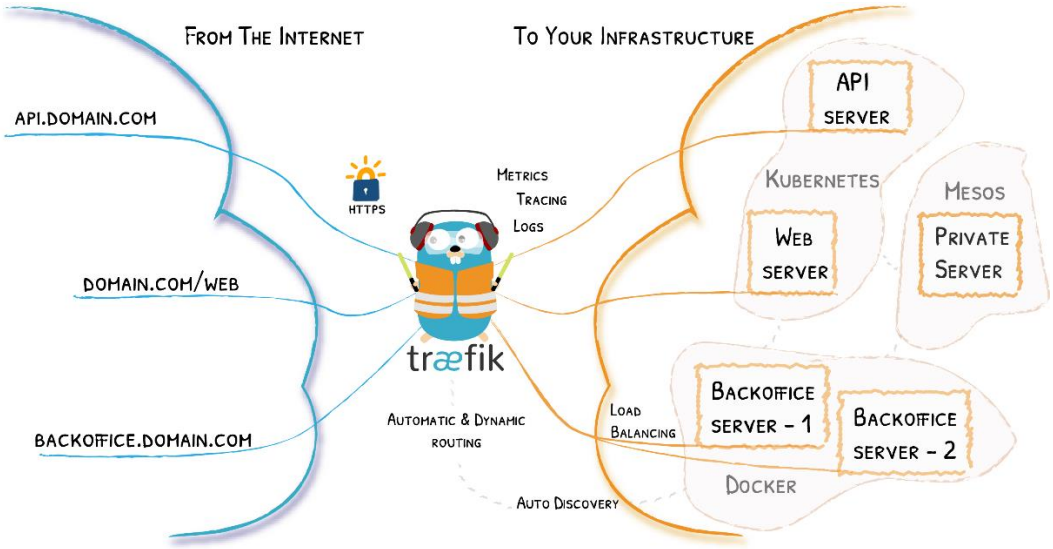


Figura 22: Funzionamento di Traefik. È in grado di associare domini che verranno esposti sul web ai corrispettivi servizi nell'infrastuttura.

3.3 Illustrazione delle Interazioni tra i Diversi Componenti

Il sistema di monitoraggio dei displacement dei Persistent Scatterers (PS) è progettato per garantire l'integrazione efficiente e accurata dei dati attraverso l'uso di microservizi orchestrati da Docker Swarm. In questa sezione, esploreremo come i vari componenti interagiscono tra loro per raccogliere, elaborare e visualizzare i dati geospaziali, migliorando la qualità e l'accuratezza dei dati generati dal processo di Persistent Scatterer Interferometry (PSI). Si presuppone che l'instradamento delle richieste e delle risposte tra i vari componenti avvenga mediante Traefik, il reverse proxy del sistema.

3.3.1 Raccolta dei Dati

Questo processo implica la raccolta e l'archiviazione delle informazioni geospaziali per garantire che i dati siano accurati e prontamente disponibili per l'analisi.

PostGIS funziona come il backend principale per la gestione dei dati spaziali all'interno del sistema. Offre un'ampia gamma di funzionalità GIS per l'archiviazione, l'analisi e la gestione dei dati relativi ai PS, compresi i dati di misurazione e le coordinate geografiche.

- **Archiviazione e Gestione dei Dati**: PostGIS estende PostgreSQL con funzionalità GIS avanzate, consentendo l'archiviazione di dati geospaziali complessi. Questo include la capacità di memorizzare dati sulle posizioni dei PS, i valori di displacement, e altri metadati associati.

- **Caricamento dei Dati**: L'utente può caricare direttamente i dati nel database in formato CSV. Questi file CSV possono contenere informazioni dettagliate sui PS, come le coordinate, le misurazioni di displacement, e altre variabili rilevanti.

3.3.2 Autenticazione e Autorizzazione

Per garantire un accesso sicuro al sistema e proteggere i dati sensibili, il processo di autenticazione e autorizzazione è gestito attraverso un'integrazione tra **OAuth2 Proxy**, **Keycloak**, **Postgres, RestAPI** e la **GUI**. Questa sezione illustra come questi componenti interagiscono per autenticare gli utenti e autorizzare le loro azioni all'interno del sistema.

❖ OAuth2 Proxy

OAuth2 Proxy funge da intermediario tra la GUI e il server di autenticazione Keycloak. Questo componente autentica gli utenti attraverso Keycloak e gestisce i token di accesso per mantenere la sessione utente.

- **Intercettazione delle Richieste**: OAuth2 Proxy intercetta le richieste degli utenti verso la GUI e reindirizza l'utente a Keycloak per l'autenticazione se non è autenticato.

- **Gestione dei Token**: Dopo aver autenticato l'utente, OAuth2 Proxy gestisce i token OAuth2 forniti da Keycloak, assicurando che l'utente abbia un token valido per accedere ai servizi protetti.

❖ Keycloak

Keycloak è il server di autenticazione che gestisce la gestione degli utenti e le autorizzazioni. Fornisce funzionalità di Single Sign-On (SSO), autenticazione e gestione dei ruoli.

- **Autenticazione**: Keycloak verifica le credenziali dell'utente e fornisce token OAuth2 validi per l'accesso. Gli utenti possono autenticarsi attraverso varie modalità, come nome utente/password o provider di identità esterni.

- **Gestione dei Ruoli**: Keycloak gestisce i ruoli e le autorizzazioni degli utenti. Gli amministratori possono definire ruoli specifici che determinano quali risorse e servizi sono accessibili agli utenti.

❖ GUI

La GUI è l'interfaccia utente che interagisce con OAuth2 Proxy e Keycloak per autenticare gli utenti e gestire l'accesso ai servizi del sistema.

- **Reindirizzamento alla Login**: La GUI reindirizza l'utente al flusso di autenticazione gestito da OAuth2 Proxy e Keycloak se l'utente non è autenticato.

- **Accesso ai Servizi**: Una volta autenticato, l'utente può accedere ai vari servizi del sistema attraverso la GUI, utilizzando i token forniti da OAuth2 Proxy per autenticare le richieste.

  Logout: La GUI fornisce una funzionalità di logout che permette all'utente di terminare la sessione corrente. Al logout, la GUI comunica con OAuth2 Proxy e Keycloak per invalidare il token OAuth2 dell'utente, terminando l'autenticazione e assicurando che il token non possa essere riutilizzato per l'accesso non autorizzato.

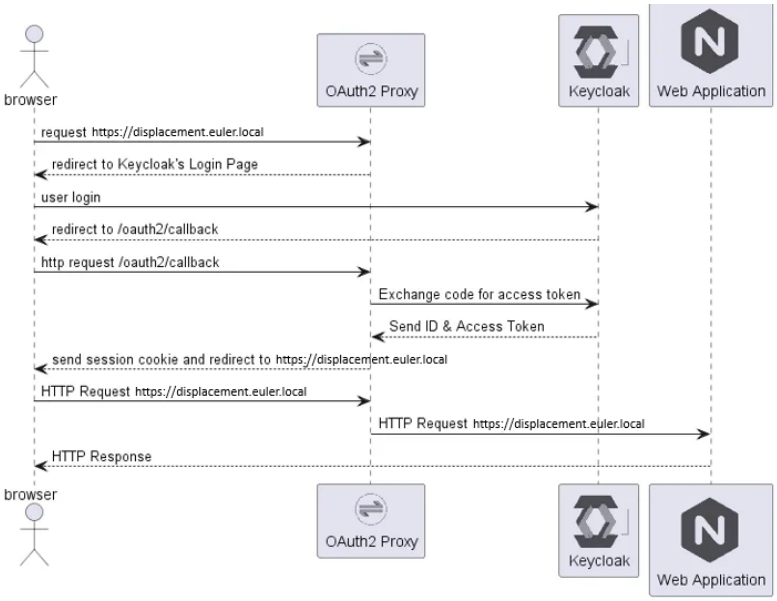


Figura 23: Diagramma di Sequenza del processo di Autenticazione.

❖ REST API

La REST API agisce come intermediario tra la GUI e GeoServer, gestendo le richieste di dati geospaziali e verificando i ruoli degli utenti tramite OAuth2 Proxy.

- **Verifica dei Ruoli**: Ad ogni richiesta ricevuta dalla GUI, la REST API verifica i ruoli dell'utente tramite OAuth2 Proxy, determinando se l'utente è autorizzato a eseguire la richiesta.

- **Accesso ai Dati**: Inoltra le richieste autorizzate a GeoServer per ottenere i dati geospaziali richiesti.

❖ Keycloak Postgres

Keycloak Postgres è il database di backend per Keycloak. Memorizza le informazioni relative agli utenti, ai ruoli, alle sessioni, e alle configurazioni di autenticazione.

- **Archiviazione dei Dati**: Memorizza i dettagli sugli utenti, inclusi i ruoli, le credenziali, e le sessioni attive.

- **Supporto per Keycloak**: Fornisce la persistenza necessaria per la gestione delle informazioni di autenticazione e autorizzazione.

3.3.3 Elaborazione e Visualizzazione dei Dati

Nella fase di elaborazione e visualizzazione dei dati, i componenti PostGIS, GeoServer, GUI, e REST API collaborano per gestire i dati dei Persistent Scatterers (PS) e fornirli in una forma utile per l'analisi e la visualizzazione. Si presuppone che l'utente abbia già completato il processo di autenticazione descritto in precedenza. Qui spieghiamo come questi componenti interagiscono per fornire funzionalità di elaborazione e visualizzazione.

❖ GeoServer

GeoServer connette al servizio PostGIS per accedere ai dati spaziali dei PS e configurare i layer e gli stili necessari per la visualizzazione.

- **Connessione a PostGIS**: Stabilisce una connessione a PostGIS, dove sono memorizzati i dati dei PS, inclusi i dati di misurazione e le coordinate geografiche.

- **Definizione di Layer e Stili**: Configura i layer geospaziali e gli stili di visualizzazione per rappresentare visivamente i dati dei PS sulla mappa. Questo include l'uso di Web Map Service (WMS) per fornire i dati mappati alla GUI.

❖ GUI

La GUI è responsabile della visualizzazione interattiva dei PS e della gestione delle richieste utente per l'accesso ai dati e le informazioni specifiche sui PS.

- **Richieste WMS a GeoServer**: La GUI effettua richieste WMS (Web Map Service) a GeoServer per visualizzare i PS sulla mappa. Questo consente di caricare e visualizzare layer mappali aggiornati dei PS direttamente nella GUI.

- **Richieste di Informazioni sui PS**: Gli utenti possono richiedere informazioni su singoli PS cliccando su di essi nella mappa. La GUI invia una richiesta alla REST API per ottenere i dettagli del PS.

- **Modulo di Filtraggio della Componente Stagionale**: La GUI include un modulo per l'analisi del time series che filtra la componente stagionale dai dati di displacement. Gli utenti possono visualizzare i risultati filtrati sulla mappa e analizzare i trend di displacement più accurati.

❖ REST API

La REST API funge da intermediario tra la GUI e GeoServer, gestendo le richieste di dati e verificando i permessi dell'utente.

- **Gestione delle Richieste**: La REST API riceve richieste dalla GUI per ottenere informazioni sui PS. Ogni richiesta include le coordinate del PS per cui si richiedono i dati.

- **Controllo dei Permessi**: La REST API verifica i permessi dell'utente tramite OAuth2 Proxy, confermando se l'utente ha il diritto di accedere ai dati richiesti. Questo controllo dei permessi segue il processo di autorizzazione già descritto.

- **Richieste a GeoServer**: Se l'utente è autorizzato, la REST API inoltra la richiesta a GeoServer per eseguire una query spaziale e ottenere i dati dettagliati del PS richiesto.

- **Restituzione dei Dati**: Riceve i dati da GeoServer e li restituisce alla GUI per la visualizzazione.

Capitolo 4

4. Implementazione della Soluzione: EULER

Questo capitolo si concentra sull'implementazione di EULER: il sistema di monitoraggio dei displacement dei Persistent Scatterers (PS). EULER prende il nome dal famoso astronomo e matematico svizzero Leonhard Euler, noto per i suoi contributi fondamentali in diverse aree della matematica e della fisica.

Verrà dettagliato il processo di implementazione pratica del sistema, con focus sui passaggi per configurare Docker Swarm e sviluppare i microservizi necessari.

4.1 Implementazione Pratica del Sistema

L'implementazione pratica del sistema ha coinvolto una serie di passaggi per configurare e mettere in funzione l'intera infrastruttura basata su Docker Swarm e i relativi microservizi. Di seguito verranno indicate le diverse scelte effettuate, le tecnologie utilizzate e le metodologie di sviluppo dell’infrastruttura e dei singoli componenti.

4.1.1 Ambiente di Sviluppo

Nel contesto della progettazione e implementazione del sistema, è stato creato un ambiente di sviluppo basato su una macchina virtuale con sistema operativo **Linux**, specificamente Ubuntu 22.04. Questa macchina virtuale è stata creata utilizzando **Vagrant**. Vagrant è uno strumento open-source per la creazione e la gestione di ambienti di sviluppo virtualizzati. Permette agli sviluppatori di definire l'infrastruttura di sviluppo come codice, utilizzando file di configurazione semplici e riproducibili.

Vagrant facilita la creazione di macchine virtuali o container su piattaforme di virtualizzazione come VirtualBox, VMware e Docker, garantendo un ambiente di sviluppo consistente e isolato.

L'uso di Linux come sistema operativo per lo sviluppo e il deploy di sistemi è una scelta comune per diverse ragioni:

- **Open Source e Flessibilità**: Linux è un sistema operativo open-source che offre una vasta gamma di distribuzioni adattabili alle esigenze specifiche dell'utente. Questa flessibilità permette di configurare e ottimizzare l'ambiente di sviluppo per adattarsi alle tecnologie e agli strumenti utilizzati nel progetto.

- **Supporto per Strumenti e Tecnologie**: Molte tecnologie di sviluppo, come Docker, Kubernetes, Python, Node.js e molti altri, sono supportate nativamente su Linux. Questo rende Linux un ambiente ideale per sviluppare e testare applicazioni che utilizzano queste tecnologie senza la necessità di emulazione o virtualizzazione aggiuntiva.

- **Stabilità e Performance**: Linux è noto per la sua stabilità e le prestazioni elevate, specialmente quando utilizzato in ambienti server. Queste caratteristiche sono cruciali per garantire un funzionamento affidabile e efficiente del sistema in produzione.

- **Sicurezza**: Linux è considerato generalmente più sicuro rispetto ad altri sistemi operativi, grazie alla sua architettura basata su permessi e alla possibilità di applicare rapidamente patch di sicurezza. Questo è particolarmente importante per i sistemi destinati all'elaborazione e alla gestione di dati sensibili, come nel caso del monitoraggio dei displacement dei Persistent Scatterers.

In sintesi, l'ambiente di sviluppo basato su Linux creato con Vagrant offre un'infrastruttura agile, riproducibile e ben integrata per la progettazione e l'implementazione del nostro sistema. Sfruttando le caratteristiche e i benefici di Linux, è possibile ottimizzare l'efficienza dello sviluppo, migliorare la sicurezza e garantire una compatibilità ottimale con le tecnologie adottate nel progetto.

4.1.2 Dataset

Il dataset utilizzato per lo sviluppo del sistema comprende 500.000 Persistent Scatterers (PS) localizzati nell'area geografica di Bari.

Ogni record nel dataset rappresenta un PS. Di seguito sono elencati i campi memorizzati per ciascuno di essi:

- **scatterer_id**: Identificativo unico per ogni PS.
- **create_date**: Data e ora di creazione del record. Impostata di default sull'ora corrente (now()).
- **update_date**: Data e ora dell'ultimo aggiornamento del record. Anche questa impostata di default sull'ora corrente.
- **coherence**: Indice di coerenza del PS. Questo valore rappresenta la qualità della misura del PS, essenziale per valutare l'affidabilità dei dati di displacement.
- **geom**: Geometria del PS memorizzata nel formato PostGIS. Questo campo contiene la rappresentazione geometrica del PS nella proiezione utilizzata.
- **height**: Altezza del PS. Questo valore rappresenta l'altitudine del PS e può essere utilizzato in analisi che richiedono la considerazione dell'elevazione.

- **lat**: Latitudine del PS. Specifica la posizione geografica del PS in termini di coordinate geografiche.
- **lon**: Longitudine del PS. Analogamente alla latitudine, specifica la posizione del PS.
- **ordering**: Campo utilizzato per l'ordinamento dei PS. Generato casualmente per garantire la distribuzione dei dati durante le operazioni.
- **geom_4326**: Geometria del PS nel sistema di riferimento EPSG:4326. Questo campo è utilizzato per operazioni e visualizzazioni che richiedono coordinate geografiche standard.
- **periodic_properties**: Proprietà periodiche del PS memorizzate in formato JSONB. Questo campo contiene informazioni aggiuntive come pattern stagionali o variazioni periodiche.
- **measurement**: Misure del PS. Questo campo contiene i dati specifici relativi ai displacement osservati.
- **velocity**: Velocità del PS. Rappresenta la velocità di spostamento del PS, misurato mm/anno. Viente utilizzata per valutare tendenze e anomalie nei dati di displacement e nella colorazione del PS nella GUI.

4.1.3 Configurazione del Docker Swarm

Durante lo sviluppo del sistema, non è stato disponibile un cluster di macchine, comunemente utilizzato per garantire sicurezza e affidabilità tramite la ridondanza dei nodi (nel caso di guasti). Essendo pertanto in un ambiente single-node, si è configurato l'ambiente di sviluppo su una singola macchina. Questo approccio semplifica la gestione e il testing dei microservizi senza la complessità aggiuntiva di un'architettura distribuita su più nodi.

Nonostante l'ambiente sia single-node, abbiamo scelto di implementare Docker Swarm anziché Docker Compose, maggiormente utilizzato in ambienti non clusterizzati. Questa decisione è motivata dalla nostra visione di espandere il sistema in futuro e di beneficiare delle capacità di gestione e scalabilità offerte da Docker Swarm, anche in un contesto di sviluppo e prototipazione. Docker Swarm, sebbene tradizionalmente utilizzato per cluster multi-nodo, offre vantaggi significativi come il load balancing dei servizi, la gestione centralizzata dei container e la facilità di scalabilità orizzontale.

Una volta preparato l'ambiente di sviluppo, abbiamo inizializzato Docker Swarm per configurare il nodo principale. Successivamente, abbiamo proceduto con la configurazione dello Swarm, definendo i tre stacks principali e le due reti overlay necessarie per gestire la comunicazione tra i microservizi.

Per garantire la sicurezza dei dati sensibili, abbiamo creato i docker secret necessari per ogni servizio, mantenendo le credenziali protette e accessibili solo ai container autorizzati.

Inoltre, per assicurare la resilienza dei dati e la disponibilità continua dei servizi, abbiamo configurato volumi Docker per ciascun servizio, adattando la capacità di storage alle esigenze specifiche di ogni componente del sistema.

Per monitorare lo stato e la salute dei servizi in esecuzione, sono stati implementati **Healthchecks**. Questi sono controlli automatici che Docker Swarm esegue per verificare che i container siano in uno stato operativo corretto. Gli Healthchecks possono verificare la risposta di un servizio a una richiesta HTTP, il corretto funzionamento di una connessione di rete o la disponibilità di risorse critiche all'interno del container.

Per configurare Docker Swarm e simulare un ambiente di produzione con supporto HTTPS, sono stati implementati certificati autofirmati per garantire la sicurezza delle comunicazioni all'interno del sistema. Sebbene in un ambiente di sviluppo l'uso di certificati autofirmati sia accettabile, per un sistema in produzione è preferibile utilizzare certificati validi emessi da autorità riconosciute. Tuttavia, per emulare le condizioni di un sistema reale, i certificati autofirmati permettono di testare la gestione delle comunicazioni sicure.

Sono stati generati certificati autofirmati per ciascun servizio chiave del sistema, con il dominio **euler.local** per simulare un ambiente di produzione. Questi certificati sono utilizzati per stabilire connessioni HTTPS, offrendo una maggiore sicurezza rispetto a HTTP non cifrato.

I certificati sono stati creati per i seguenti domini:

euler.local: Utilizzato da Traefik per l’esposizione della dashboard.

displacement.euler.local: Utilizzato dalla GUI per garantire che le comunicazioni tra l'interfaccia utente e il sistema siano crittografate.

keycloak.euler.local: Utilizzato da Keycloak per gestire l'autenticazione degli utenti in modo sicuro.

geoserver.euler.local: Utilizzato da GeoServer per fornire i dati geospaziali in modo sicuro alle richieste provenienti dagli altri componenti del sistema.

Queste configurazioni hanno fornito una base solida per l'implementazione e il deployment del sistema EULER, garantendo la sicurezza, l'affidabilità e la gestione efficiente delle risorse durante lo sviluppo e il testing delle funzionalità di monitoraggio dei displacement dei Persistent Scatterers (PS).

4.1.4 Sviluppo dei Microservizi

In questa sezione verrà illustrato nello specifico lo sviluppo dei microservizi che compongono il sistema EULER.

4.1.4.1 GUI

Il servizio GUI è stato sviluppato utilizzando **Flask**, un framework leggero per **Python** progettato per costruire applicazioni web. Flask facilita la gestione delle richieste HTTP e la creazione di API per interagire con altre componenti del sistema. Questo framework è scelto per la sua semplicità e flessibilità, adatto per progetti di dimensioni variabili.

La parte frontend è stata implementata principalmente utilizzando HTML, JavaScript e CSS per garantire un'interfaccia utente reattiva e intuitiva.

Il servizio è composto da due moduli:

- **Modulo Principale (app.py)**: Questo modulo funge da backend per l'applicazione GUI. Utilizza Flask per gestire le richieste HTTP, elaborare i dati provenienti dall'interfaccia utente e fornire risposte appropriate. Le funzioni definite in app.py sono chiamate dalle richieste JavaScript per eseguire operazioni come il recupero dei dati dai microservizi sottostanti e l'aggiornamento dell'interfaccia utente con i risultati ottenuti.
- **Modulo per l'Analisi Stagionale (stagionality_analysis.py)**: Questo modulo è dedicato all'analisi dei dati stagionali dei Persistent Scatterers (PS). Implementa algoritmi e logiche specifiche per filtrare le fluttuazioni stagionali dai dati di displacement, migliorando così l'accuratezza delle analisi.

4.1.4.1.1 Implementazione del Frontend

Il frontend dell'applicazione GUI è stato implementato utilizzando HTML e JavaScript (JS), con l'obiettivo di fornire all'utente un'interfaccia interattiva e intuitiva per interagire con il sistema di monitoraggio dei displacement dei Persistent Scatterers (PS). Ecco come è strutturato e implementato il frontend:

❖ Componenti del Frontend

- **Controller del PS**: Gestisce l'interazione dell'utente con i Persistent Scatterers (PS). Include funzionalità per visualizzare i PS sulla mappa, interagire con le loro informazioni e avviare operazioni specifiche su di essi.

- **Controller della Mappa**: Utilizza Leaflet, una libreria JavaScript open-source per mappe interattive, per visualizzare e gestire la mappa. Integra le funzionalità di zoom, panoramica e overlay dei PS sulla mappa. Gestisce gli eventi di click e hover sulla mappa per fornire all'utente una navigazione fluida e dettagliata dei dati geospaziali.

- **Controller dell'Utente**: Gestisce le operazioni di autenticazione utente, inclusi i flussi di login e logout. Utilizza le API fornite da OAuth2Proxy e Keycloak per garantire l'accesso sicuro alle funzionalità dell'applicazione.

❖ Interazione con la Mappa e i Persistent Scatterers

- **Visualizzazione dei PS sulla Mappa**: I PS sono visualizzati sulla mappa utilizzando una richiesta Web Map Service (WMS) a GeoServer. Leaflet è configurato per mostrare i dati geospaziali forniti da GeoServer, consentendo agli utenti di esplorare i PS in diversi contesti spaziali. I PS avranno colore diverso a seconda della velocità di displacement annua, come è possibile osservare nella legenda posta in basso a sinistra.

Figura 24: Schermata del servizio GUI. Mostra i diversi PS nella mappa nella sezione geografica di Bari.

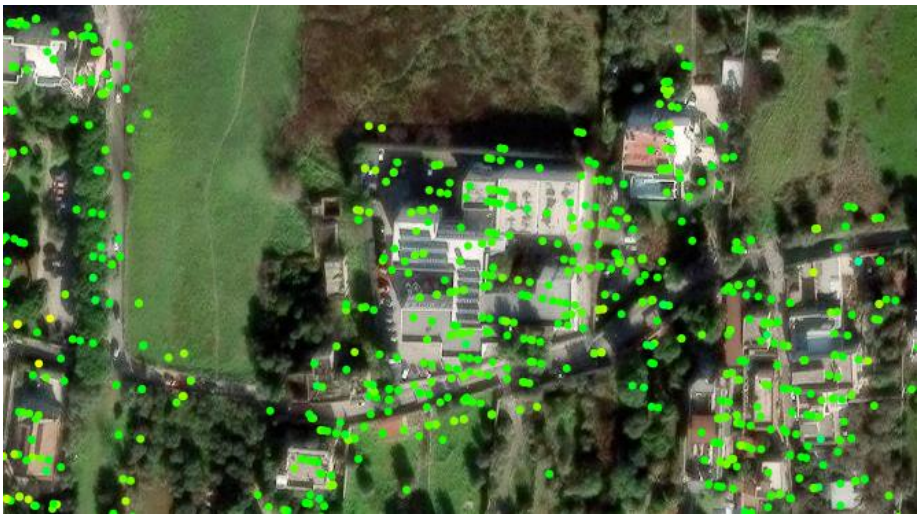


Figura 25: Zoom su una zona di Bari per osservare meglio i singoli PS. È pertanto possibile osservare come i PS individuati si trovino su edifici e strade, mentre non vengono individuati sulla vegetazione.

- **Dettagli dei PS**: Cliccando su un singolo PS sulla mappa, l'utente può accedere a informazioni dettagliate. Le informazioni sul PS, comprese le sue coordinate geografiche e i dati di displacement, vengono visualizzate in un grafico interattivo. Il grafico è costruito utilizzando Chart.js, una libreria JavaScript per la creazione di grafici dinamici e interattivi.

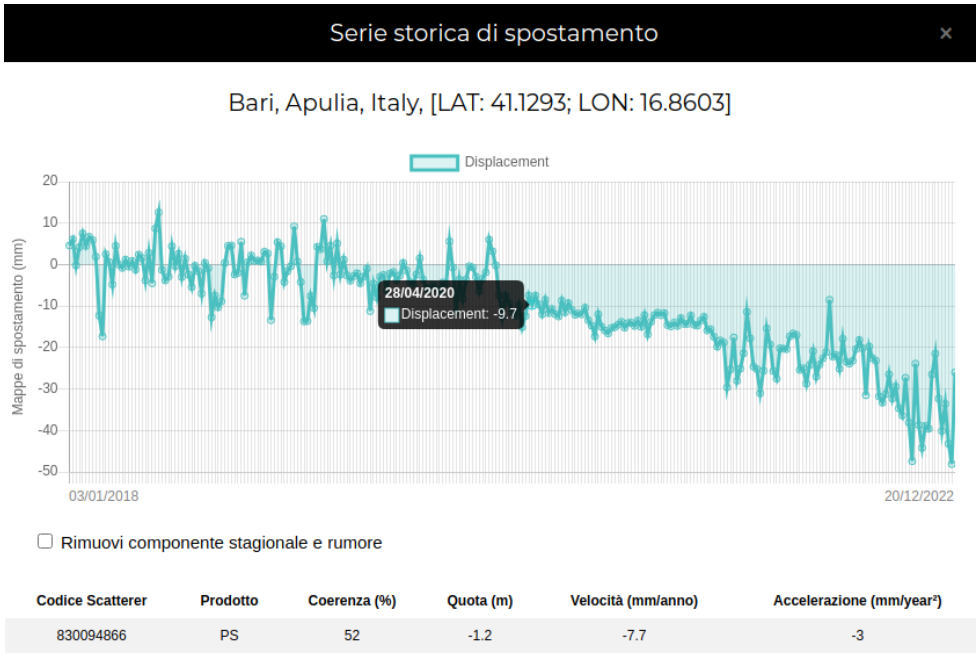


Figura 26: Displacement del PS e informazione su di esso.

- **Rimozione della Componente Stagionale**: L'interfaccia utente include un filtro per la rimozione delle fluttuazioni stagionali dai dati di displacement. Cliccando sul filtro, l'utente può visualizzare una versione depurata dei dati di displacement che esclude le variazioni dovute a fattori stagionali e il rumore. Il modulo per l'analisi stagionale gestisce l'applicazione di algoritmi per filtrare e visualizzare i dati corretti, migliorando la precisione dell'analisi.

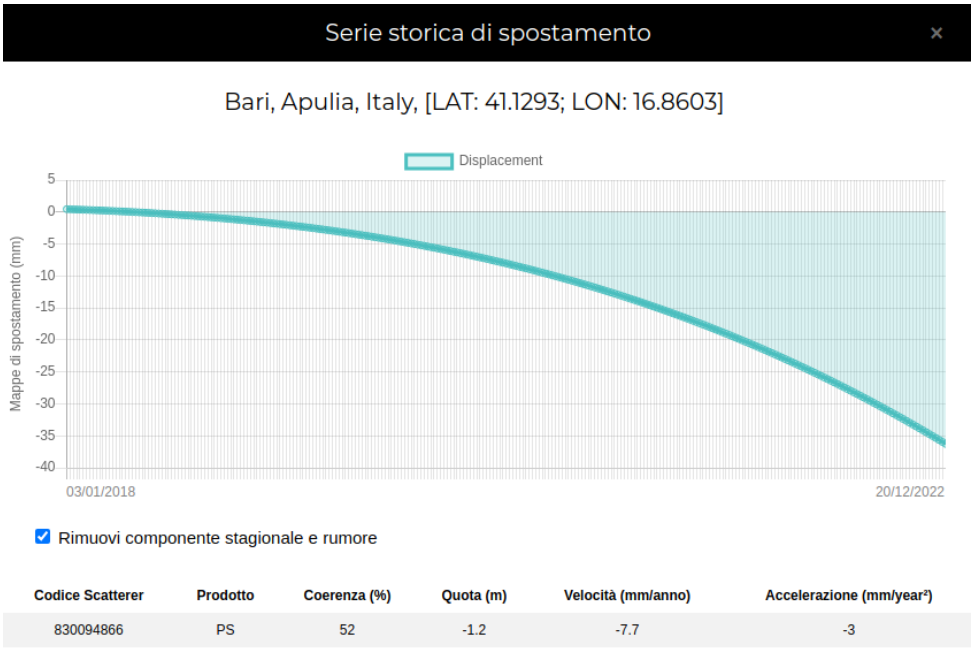


Figura 27: Displacement del PS con rimozione della componente stagionale e rumore.

4.1.4.1.2 Implementazione della Rimozione della Componente Stagionale

In questa sezione, verrà illustrata l'implementazione della rimozione della componente stagionale dai dati di displacement dei Persistent Scatterers per ottenere una rappresentazione accurata del trend sottostante. Questo processo permette di isolare i veri movimenti di terreno dai cambiamenti dovuti a variazioni stagionali e rumore. L'algoritmo utilizzato per questa operazione è basato sull'implementazione end-to-end proposta dal servizio European Ground Motion Service (EGMS) del programma Copernicus [6].

❖ Descrizione Generale dell'Algoritmo

Consideriamo che la serie temporale dei dati di displacement $y(t)$ possa essere espressa come una combinazione di tre componenti principali:

$$y(t) = Trend(t) + Stagionalità(t) + Rumore(t)$$

L'obiettivo è rimuovere la componente stagionale e il rumore per ottenere il trend:

$$Trend(t) = y(t) - Stagionalità(t) - Rumore(t)$$

❖ Scelta del Modello

1. **Trend** (Polinomio di Terzo Grado):

Il trend è modellato come un polinomio di terzo grado:

$$\text{Trend}(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

Questo modello permette di catturare curvature complesse e cambiamenti nella tendenza dei dati, fornendo una maggiore flessibilità rispetto a un modello lineare o quadratico.

2. **Componente Stagionale** (Funzioni Sinusoidali):

La componente stagionale è rappresentata da una combinazione di sinusoidi:

$$\text{Stagionalità}(t) = b_1 \cos(2\pi t) + b_2 \sin(2\pi t)$$

Le funzioni sinusoidali sono utilizzate per modellare variazioni periodiche che si ripetono nel tempo, come le fluttuazioni annuali o semestrali.

3. Rumore:

Il rumore $Rumore(t)$ rappresenta le variazioni casuali e gli errori di misura. È identificato come il residuo rimanente dopo la sottrazione delle componenti di trend e stagionalità dai dati osservati.

❖ Costruzione del Modello

1. Definizione del Tempo:

Il tempo t rappresenta l'indice temporale della serie temporale dei dati. Si è considerato un periodo di campionamento unitario.

$$t = \{0, 1, 2, \dots, n-1\}$$

2. Matrice Modello G:

La matrice modello G combina i termini del polinomio di terzo grado e le componenti sinusoidali:

$$G = [\ \ t\text{^}3 \ \ t\text{^}2 \ \ t \ \ 1 \ \ \cos(2\,\pi t) \ \ \sin(2\,\pi t)\ ]$$

3. Calcolo dei Coefficienti:

I coefficienti del modello $\boldsymbol{COEFF}$ sono calcolati utilizzando la pseudo-inversa della matrice modello G e la serie temporale osservata $\boldsymbol{y}$.

$\boldsymbol{COEFF} = (G^T G)^{-1} G^T y$

4. Calcolo del Modello:

Il modello stimato \(\mathbf{M} \) approssima la serie temporale osservata \(\mathbf{y} \):

$$M = G\ COEFF$$

5. Calcolo della Componente Stagionale:

La componente stagionale $Stagionalità(t)$ viene estratta utilizzando i coefficienti sinusoidali:

$$\text{Stagionalità}(t) = b_1 \cos(2\pi t) + b_2 \sin(2\pi t)$$

6. Calcolo del Residuo:

Il residuo $Rumore(t)$ è la differenza tra la serie temporale osservata $\boldsymbol{y}$ e il modello stimato $\boldsymbol{M}$:

$$\text{Rumore}(t) = y - M$$

7. Isolamento del Trend:

Il trend $Trend(t)$ viene ottenuto sottraendo la componente stagionale e il rumore dai dati osservati:

$$\text{Trend}(t) = y(t) - \text{Stagionalità}(t) - \text{Rumore}(t)$$

4.1.4.2 REST API

l servizio REST API, sviluppato in Java Spring, fornisce l'accesso ai dati geospaziali e supporta le operazioni richieste dal sistema. Java Spring è un framework open-source per il linguaggio Java, progettato per semplificare lo sviluppo di applicazioni aziendali e web. Offre un'architettura modulare che promuove una facile integrazione di varie funzionalità, come la gestione delle richieste HTTP, la configurazione dei componenti, e la sicurezza. Spring consente lo sviluppo rapido, sicuro e scalabile di applicazioni Java attraverso l'iniezione di dipendenze e la programmazione orientata agli aspetti.

l modulo REST API è strutturato in tre componenti principali:

1. FeaturesController

Il controller FeaturesController gestisce le richieste HTTP per ottenere le informazioni sui feature geospaziali. Implementa un endpoint che riceve i parametri di bounding box (latitudine e longitudine minime e massime) e un cookie di autenticazione per determinare se l'utente ha i permessi necessari per accedere alle informazioni richieste.

2. CookieValidation

La classe CookieValidation gestisce la validazione del cookie di autenticazione. Verifica se l'utente appartiene al gruppo di utenti con permessi speciali ("/SPECIALIST") necessari per accedere alle informazioni richieste.

3. HttpInsecureRequest

La classe HttpInsecureRequest crea un client HTTP che può inviare richieste a server con certificati SSL autofirmati. Questo è necessario poiché il sistema utilizza certificati autofirmati per le comunicazioni interne.

4.1.4.3 Geoserver

GeoServer è stato deployato utilizzando l'immagine Docker di **Kartoza**, nota per fornire una versione preconfigurata di GeoServer che facilita la gestione e la distribuzione di dati geospaziali. Questa configurazione permette di integrare facilmente GeoServer con il sistema di monitoraggio EULER per fornire servizi di visualizzazione e accesso ai dati geospaziali.

La configurazione di GeoServer è stata suddivisa in diverse fasi:

1. **Connessione al Database PostGIS**: Nella configurazione del Data Store sono stati specificati i parametri necessari, come l'URL del database, il nome utente e la password per l'autenticazione, nonché il driver JDBC per la connessione a PostgreSQL.

2. **Definizione del Layer**: Un Layer è stato definito in GeoServer per rappresentare i PS. Questo layer è basato sul Data Store precedentemente configurato e utilizza la tabella ps_measurements del database PostGIS.

Il layer è stato configurato con un Bounding Box (BBOX) che definisce i confini spaziali dei dati da visualizzare. Il sistema di riferimento spaziale (SRS) utilizzato è l'EPSG:4326, che è uno standard comunemente utilizzato per i dati geografici.

3. **Definizione degli Stili**: Gli stili sono stati creati utilizzando il SLD (Styled Layer Descriptor) per definire come i PS devono essere rappresentati sulla mappa. In particolare, la colorazione dei PS varia in base alla velocità di spostamento annuo, consentendo una visualizzazione intuitiva delle aree con diversi livelli di displacement.

4. **Esposizione del Layer**: Il layer è stato esposto tramite il servizio WMS (Web Map Service) di GeoServer, permettendo alle applicazioni client di richiedere mappe geospaziali e visualizzare i PS sulla mappa.

4.1.4.4 OAuth2proxy

L’OAuth2 Proxy è stato configurato per gestire l’autenticazione degli utenti tramite il protocollo OAuth 2.0, utilizzando Keycloak come provider di identità. Questa configurazione consente agli utenti di autenticarsi e accedere al servizio GUI di EULER, integrando un livello di sicurezza per la gestione delle credenziali e il controllo degli accessi.

La configurazione è avvenuta mediante l'impostazione delle **variabili d'ambiente** per definire l'upstream target della GUI di EULER, specificare Keycloak come provider OAuth 2.0 con il protocollo OpenID Connect, stabilire il tempo di scadenza dei cookie di sessione, passare l'access token al backend, definire l'URL di callback per la redirezione dopo il login, definire gli URL di login, riscatto dell'access token, chiavi di firma JWT, e logout, attivare la protezione CSRF per ogni richiesta, e autorizzare i domini per il login.

4.1.4.5 Keycloak

Keycloak è stato configurato mediante la creazione di un nuovo realm dedicato al sistema EULER. All'interno del realm sono stati configurati un client specifico per EULER, ruoli personalizzati per gestire l'accesso e le autorizzazioni degli utenti, e utenti con i relativi dati di accesso. Sono stati impostati gli URL di callback per la gestione del login, del logout e altre operazioni relative all'OAuth2 e OpenID Connect. Inoltre, è stata effettuata la connessione a un database PostgreSQL per memorizzare in modo sicuro le informazioni degli utenti, inclusi ruoli e altre informazioni pertinenti.

4.1.4.6 Traefik Proxy

Traefik Proxy è stato configurato in modo da gestire il routing e l'endpoint HTTPS per tutti i servizi all'interno dell'ambiente EULER. Questa configurazione è essenziale per garantire una comunicazione sicura attraverso HTTPS, utilizzando certificati SSL/TLS autofirmati per i domini specificati come displacement.euler.local, keycloak.euler.local, e geoserver.euler.local. Tali certificati sono stati inseriti all’interno del container di Traefik mediante volumi.

4.1.5 Deployment e Gestione delle Versioni

Per automatizzare il deployment e gestire le versioni del sistema EULER, sono stati adottati diversi approcci e strumenti:

- Deployment Automatizzato

Dopo la configurazione iniziale del sistema, che include la creazione di Docker Swarm, dei secret e dei volumi necessari, il processo di deployment è strutturato nel seguente modo:

Costruzione dei Docker-Compose dagli Stack: Gli stack Docker sono definiti per ciascun servizio (come GUI, Geoserver, OAuth2proxy) usando file YAML che descrivono i servizi, le reti e le configurazioni necessarie.

Deploy degli Stack: Una volta che i Docker-Compose sono stati costruiti, è possibile deployare gli stack all'interno del Docker Swarm. Questo passaggio viene eseguito per avviare e gestire i servizi all'interno dell'ambiente di sviluppo.

Configurazione degli /etc/hosts: Poiché i servizi sono accessibili tramite domini definiti localmente come displacement.euler.local, keycloak.euler.local e geoserver.euler.local, è necessario configurare gli indirizzi IP corrispondenti nei file /etc/hosts del sistema operativo. Questo consente la risoluzione dei DNS locali per i servizi durante lo sviluppo e il testing.

Configurazione del Browser: Per garantire che il browser riconosca i certificati SSL/TLS autofirmati utilizzati per i domini euler.local, è necessario importare manualmente questi certificati nel browser.

- ❖ Gestione delle Versioni con GitHub

Repository GitHub: È stato creato un repository su GitHub come spazio progettuale centrale per il codice sorgente e la gestione delle versioni del progetto EULER. GitHub fornisce un ambiente collaborativo per sviluppare, testare e rilasciare il software in modo organizzato.

Metodologia delle Feature: Per lo sviluppo in ambiente di sviluppo (development), è stato utilizzato il modello delle feature. Questo approccio organizza lo sviluppo in base a singole funzionalità o miglioramenti, gestendo le modifiche come branch separati nel repository. Ogni feature branch viene sviluppato, testato e integrato indipendentemente, garantendo una migliore gestione del codice e facilitando il rilascio incrementale delle nuove funzionalità.

Capitolo 5

5. Applicazioni e prospettive future

l monitoraggio preciso dello spostamento del suolo fino al millimetro riveste un'importanza determinante in diversi ambiti applicativi. Questo tipo di monitoraggio può essere utilizzato per:

Monitoraggio e Valutazione dei Disastri Naturali: È fondamentale per osservare e valutare le conseguenze di eventi come terremoti e frane. Il rilevamento preciso dei movimenti del suolo consente di comprendere meglio l'evoluzione delle deformazioni post-evento e valutare i rischi residui.

Monitoraggio dei Sistemi di Infrastrutture Critiche: Per esempio, il monitoraggio dei tubi fognari può rilevare perdite d'acqua attraverso il movimento del suolo, che può gonfiarsi a causa dell'acqua intrappolata, segnalando potenziali problemi di infrastruttura prima che diventino critici.

Impatti delle Condizioni Ambientali: Il monitoraggio dei movimenti del suolo può essere correlato alle variazioni climatiche. Ad esempio, materiali come il metallo tendono a dilatarsi con alte temperature, mentre terreni ghiaiosi possono sollevarsi durante periodi di piogge frequenti.

Monitoraggio del Livello del Mare: L'osservazione del movimento del suolo può aiutare a prevedere gli effetti di un aumento del livello del mare sulle barriere costiere e sulle aree vulnerabili, fornendo informazioni cruciali per la pianificazione e la mitigazione dei rischi.

Nel contesto delle prospettive future per il monitoraggio dei Persistent Scatterers (PS) e dello spostamento del suolo, si delineano sviluppi significativi che puntano alla avanzata e alla diversificazione delle tecnologie impiegate. Un obiettivo primario è rappresentato dall'adozione di tecnologie di rilevamento sempre più avanzate e precise con capacità elevate di risoluzione spaziale e temporale. Questo approccio mira non solo a migliorare la precisione delle misurazioni, ma anche a incrementare la frequenza di acquisizione dei dati, consentendo una monitoraggio più dettagliato e reattivo dei movimenti del suolo nel tempo.

Parallelamente, si prospetta un'integrazione più profonda di dati provenienti da diverse fonti sensoriali, tra cui SAR, GPS e dati meteorologici. Questa integrazione mira a fornire una visione integrata e multidimensionale dei fenomeni di spostamento del suolo, consentendo una comprensione più completa e contestualizzata degli effetti delle variazioni ambientali e delle attività umane su tali fenomeni.

Lo sviluppo di modelli predittivi avanzati costituisce un ulteriore ambito di evoluzione. Utilizzando tecniche di machine learning e analisi avanzata dei dati, si mira a sviluppare modelli capaci di anticipare e gestire i cambiamenti nel suolo con maggiore accuratezza e tempestività. Questi modelli non solo supportano la pianificazione e la gestione del territorio, ma anche la previsione di rischi associati a eventi naturali e attività umane.

Infine, l'applicazione del monitoraggio dei movimenti del suolo in settori emergenti come l'urbanistica intelligente, la gestione delle risorse idriche e la pianificazione del territorio rappresenta un'altra prospettiva chiave. Questi settori possono beneficiare del monitoraggio continuo e dettagliato dei PS per migliorare la resilienza delle infrastrutture e promuovere la sostenibilità ambientale, affrontando sfide come il cambiamento climatico e la crescita urbana.

Conclusioni

Nel complesso, il progetto di sviluppo di EULER, dedicato al monitoraggio dei Persistent Scatterers (PS) e dello spostamento del suolo, rappresenta un passo significativo verso l'implementazione di soluzioni avanzate nel campo della geomatica e della monitoraggio ambientale. Attraverso l'integrazione di tecnologie moderne e l'adozione di approcci innovativi, il sistema è stato progettato per fornire una piattaforma robusta e scalabile per analizzare e gestire i cambiamenti del suolo con elevata precisione e affidabilità.

Durante l'implementazione, sono state affrontate diverse sfide, tra cui la configurazione di un ambiente Docker Swarm per garantire la sicurezza e la scalabilità del sistema anche in un contesto single-node. L'adozione di Docker Swarm anziché Docker Compose, pur in un ambiente single-node, ha posto le basi per future espansioni e miglioramenti, evidenziando un approccio proattivo alla gestione delle risorse e delle operazioni di deployment.

La componente software del sistema è stata sviluppata utilizzando diverse tecnologie moderne, come Java Spring per le REST API, Flask in Python per l'interfaccia utente, e l'integrazione di componenti come Geoserver per la gestione dei dati geospaziali e Keycloak per l'autenticazione e l'autorizzazione degli utenti. Questa diversità tecnologica non solo ha ampliato le capacità funzionali di EULER, ma ha anche creato una base solida per futuri miglioramenti e aggiornamenti.

L'implementazione di un sistema di autenticazione basato su OAuth2 con Keycloak e l'integrazione di Traefik Proxy per la gestione delle route e la sicurezza dei servizi hanno ulteriormente potenziato la sicurezza e l'affidabilità del sistema, assicurando al contempo un accesso controllato e sicuro alle risorse.

Infine, guardando al futuro, le prospettive per il progetto EULER includono l'adozione di tecnologie avanzate di rilevamento, l'integrazione di dati multipli per una visione più completa dei movimenti del suolo, lo sviluppo di modelli predittivi avanzati e l'esplorazione di nuove applicazioni in settori emergenti. Questi sviluppi mirano a migliorare ulteriormente la capacità di EULER di supportare la pianificazione urbana, la gestione delle risorse naturali e la sostenibilità ambientale in risposta alle sfide globali.

In conclusione, il progetto EULER rappresenta un esempio tangibile di come l'innovazione tecnologica possa essere applicata con successo per affrontare questioni cruciali legate alla monitoraggio del suolo e alla gestione ambientale, con benefici significativi per comunità, infrastrutture e ambiente.

Bibliografia

[1] A. M. H. Ansar, et al. “A SHORT REVIEW on PERSISTENT SCATTERER INTERFEROMETRY TECHNIQUES for SURFACE DEFORMATION MONITORING.” ˜ *the œInternational Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences/International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVI-4/W3-2021, 10 Jan. 2022, pp. 23–31, https://doi.org/10.5194/isprs-archives-xlvi-4-w3-2021-23-2022. Accessed 17 Apr. 2024.

[2] “About - GeoServer.” *Geoserver.org*, geoserver.org/about/.

[3] “Che Cos’è Il Radar, Come Funziona E Come è Fatto.” *Geopop*, Geopop, 2022, www.geopop.it/che-cose-come-e-fatto-e-come-funziona-il-radar/. Accessed 6 June 2024.

4[4] “Cos’è Un’API REST? | IBM.” *Www.ibm.com*, 3 Apr. 2024, www.ibm.com/it-it/topics/rest-apis. Accessed 21 June 2024.

[5] Earth Science Data Systems, NASA. “Earthdata.” *Earthdata*, 2019, www.earthdata.nasa.gov/learn/backgrounders/what-is-sar.

[6] “End-To-End Implementation and Operation of the European Ground Motion Service (EGMS).” *Https://Land.copernicus.eu/En*, Copernicus, Oct. 25AD.

[7] “European Space Agency.” *Www.esa.int*, www.esa.int/SPECIALS/Eduspace_Global_IT/SEMLT0G64RH_0.html. Accessed 6 June 2024.

[8] Evers, M., et al. “CONCEPT to ANALYZE the DISPLACEMENT TIME SERIES of INDIVIDUAL PERSISTENT SCATTERERS.” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B3-2021, 28 June 2021, pp. 147–154, https://doi.org/10.5194/isprs-archives-xliii-b3-2021-147-2021. Accessed 16 Jan. 2023.

[9] Ferretti, Alessandro, et al. *InSAR Principles-Guidelines for SAR Interferometry Processing and Interpretation*. Vol. 19, 1 Feb. 2007. Accessed 8 June 2024.

[10] “GeoServer — OSGeoLive 16.0 Documentation.” *Live.osgeo.org*, live.osgeo.org/it/overview/geoserver_overview.html. Accessed 21 June 2024.

[11] Hajnsek, Irena, and Yves-Louis Desnos. *Polarimetric Synthetic Aperture Radar : Principles and Application*. Cham, Switzerland, Springer, 2020.

[12] “InSAR Interferometric Synthetic Aperture Radar - TRE ALTAMIRA.” *TRE ALTAMIRA - a CLS Group Company*, site.tre-altamira.com/insar/.

[13] “Keycloak.” *Www.keycloak.org*, www.keycloak.org.

[14] Podest, Erika . “Basics of Synthetic Aperture Radar.”

[15] PostGIS Developers. “PostGIS — Spatial and Geographic Objects for PostgreSQL.” *Postgis.net*, 2020, postgis.net/.

[16] Sancho, Candela. “PS-InSAR & DS-InSAR Algorithms: Permanent and Distributed Scatterers.” *Detektia Earth Surface Monitoring*, 25 Apr. 2023, detektia.com/en/insar/ps-insar-ds-insar/#PS-InSAR_Persistent_Scatterer_Interferometry. Accessed 11 June 2024.

[17] “Spettro Elettromagnetico | Descrizione E Caratteristiche.” *Electricity - Magnetism*, 14 Jan. 2024, www.electricity-magnetism.org/it/spettro-elettromagnetico-descrizione-e-caratteristiche/. Accessed 6 June 2024.

[18] “STRATEGIE DI CLASSIFICAZIONE AUTOMATICA DI IMMAGINI MULTISPETTRALI per AREE URBANE.” *SlideShare*, 29 Jan. 2016, www.slideshare.net/slideshow/strategie-di-classificazione-automatica-di-immagini-multispettrali-per-aree-urbane/57650071#15. Accessed 7 June 2024.

[19] Ulaby, F. T, et al. *Microwave Remote Sensing: Active and Passive*. Wiley. Accessed 2014.

[20] “What Is OAuth 2.0 and What Does It Do for You?” *Auth0*, auth0.com/intro-to-iam/what-is-oauth-2.