

Ricerca di Sistema elettrico



LA2.14 CI-RES: Definizione della piattaforma di servizi e verifica dei servizi di back-end

Gregorio D'Agostino, Antonio Di Pietro, Maurizio Pollino, Alberto Tofani

LA2.14 CI-RES: DEFINIZIONE DELLA PIATTAFORMA DI SERVIZI E VERIFICA DEI SERVIZI DI BACK-END

Gregorio D'Agostino, Antonio Di Pietro, Maurizio Pollino, Alberto Tofani (ENEA)

Giugno 2023

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA
Piano Triennale di Realizzazione 2022-2024

Obiettivo: *Decarbonizzazione/Digitalizzazione ed evoluzione delle reti*

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Linea di attività: 2.14

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile Linea di Attività: Gregorio D'Agostino - ENEA

Mese inizio previsto: 1

Mese inizio effettivo: 18

Mese fine previsto: 1

Mese fine effettivo: 18

Indice

1	RISULTATI ATTESI	3
2	RISULTATI OTTENUTI.....	4
3	PRODOTTI ATTESI.....	5
4	PRODOTTI SVILUPPATI	6
5	ANALISI DEGLI SCOSTAMENTI SU ATTIVITÀ E RISULTATI	7
6	SINTESI DELLE ATTIVITÀ SVOLTE	8
7	DETTAGLIO DELLE ATTIVITÀ SVOLTE.....	9
8	CONTRIBUTO DELLE EVENTUALI CONSULENZE ALLE ATTIVITÀ SOPRA DESCRITTE.....	15
9	PUBBLICAZIONI SCIENTIFICHE.....	16
10	EVENTI DI DISSEMINAZIONE	17

1 Risultati attesi

L'obiettivo della linea di attività LA 2.14 è la pianificazione della infrastruttura di servizio CI-RES (Critical Infrastructure Resilience) e della definizione delle sue funzionalità operative, delle sue finalità e nella formalizzazione delle specifiche tecniche. L'architettura modulare (a micro/meso-servizi) prevede la definizione di un'architettura a più livelli dove si distinguono: 1) i servizi di calcolo con la definizione dei diversi workflow operativi e algoritmi, questo livello è detto anche di 'back-end'; 2) i servizi di gestione ed interrogazione delle basi di dati (livello dati) e 3) il supporto delle interfacce per l'utente, il cosiddetto livello di 'front-end'.

Per quanto riguarda l'integrazione di dati da UAS (Unmanned Aerial Systems), sono state definite le categorie di dati che saranno integrate nella piattaforma a supporto del monitoraggio delle Infrastrutture.

L'obiettivo generale che si vuole ottenere è l'ottimizzazione dei servizi di back-end in termini di prestazione, modularità e fruibilità. Molti servizi rappresenteranno una re-ingegnerizzazione e un miglioramento degli esistenti, altri saranno il risultato di una integrazione sinergica e altri ancora saranno realizzati ex-novo.

2 Risultati ottenuti

Il risultato principale ottenuto, formalizzato in un documento di specifica, è stato un'attenta analisi dei requisiti funzionali e non funzionali della piattaforma CI-RES. Molto utili, al fine del raggiungimento del risultato stesso, sono stati i diversi meeting tenutisi con aziende specializzate in sviluppo piattaforme individuate come potenziali realizzatrici. Il documento di specifica è quindi il frutto di iterazioni continue tramite le quali sono stati discussi i requisiti dei vari componenti dell'architettura, le possibili soluzioni tecnologiche per rispondere a tali requisiti insieme ad un'attenta analisi costo benefici di ogni singola scelta. Le categorie di dati provenienti da UAS, a supporto del monitoraggio delle Infrastrutture sono elencate di seguito per quanto riguarda le infrastrutture di trasporto e quelle di rete. Le infrastrutture di trasporto (stradali, ferroviarie, aeroportuali, portuali, idroviarie) necessitano di periodici interventi di verifica al fine di verificare lo stato e dunque programmare gli interventi utili a fornire una continuità di esercizio e la stabilità. L'utilizzo degli UAS permette di ottenere:

1. Immagini ottenute da tecniche fotogrammetriche utili per produrre modelli digitali 2D e 3D misurabili;
2. Dati ottenuti da rilievi con sensore LiDAR L1 per ottenere modelli DSM (Digital Surface Model) o DTM (Digital Elevation Model) anche al di sotto della copertura vegetativa;
3. Immagini ottenute con fotocamere RGB dotate di uno zoom molto efficace (es. DJI Zenmuse Z30), tali da consentire l'acquisizione d'immagini dettagliate anche di particolari posizionati in zone inaccessibili;
4. Immagini ottenute da camere termiche radiometriche (es. FLIR 640x512 @30Hz / R) che permettono l'identificazione di zone soggette ad infiltrazioni e ammaloramento.

Le infrastrutture di rete basate su distribuzione capillare includono i sistemi di distribuzione elettrica, dell'acqua, i sistemi di telecomunicazione. Anche questi sistemi necessitano continua manutenzione e controlli periodici su tratti molto estesi. In particolare, per quanto riguarda le reti elettriche (media, alta ed altissima tensione), le ispezioni consentono di ottenere le seguenti tipologie di dati:

1. modello digitale ottenuto tramite tecniche aerofotogrammetriche per la verifica di eventuali problemi strutturali (difetti di assemblaggio, presenza di cricche, deformazioni, ossidazioni, etc.);
2. Immagini con fotocamere RGB in alta risoluzione dotate di zoom ottico 30x, per la verifica di dettaglio degli isolatori, degli ambari, dei cimini e delle mensole, utile per indagare l'esistenza di eventuali difetti, rotture o ammaloramenti;
3. Immagini termografiche degli isolatori e dei conduttori utile per rilevare gli eventuali difetti di conduzione elettrica;
4. Dati ottenuti tramite l'utilizzo di sensori LIDAR per il rilievo e analisi dei conduttori aerei;

Per quanto riguarda le reti di distribuzione dell'acqua e del gas, essendo queste reti quasi interamente interrate, non risultano ispezionabile con gli UAS. Ad ogni modo è possibile utilizzare gli UAS per effettuare rilievi fotografici atti a effettuare controlli visivi su dighe e tubazioni a cielo aperto, oppure per realizzare rilievi morfologici del territorio, utili ai fini della progettazione di ampliamenti e/o per la messa in sicurezza di tratti interessati da dissesto.

3 Prodotti attesi

Il prodotto atteso è un rapporto tecnico che descrive l'architettura della nuova piattaforma CI-RES e i requisiti funzionali e strutturali della stessa. I requisiti saranno rappresentati utilizzando opportuni formati standard in maniera tale da garantirne il rispetto nella realizzazione della infrastruttura e il tracciamento di eventuali varianti ed estensioni nelle attività future. I requisiti saranno organizzati e presentati secondo uno schema di rilevanza: *“must, should e could”*.

4 Prodotti sviluppati

Il prodotto principale della presente linea di attività consiste nel presente rapporto tecnico “CI-RES: definizione della piattaforma di servizi e verifica dei servizi di back-end” che descrive le funzionalità della piattaforma e le sue specifiche.

I requisiti sono descritti sinteticamente nelle tabelle contenute nella sezione 7 del presente rapporto tecnico. Tali tabelle descrivono, utilizzando uno specifico standard, i requisiti funzionali e strutturali della piattaforma CI-RES. L’analisi della complessità di realizzazione delle applicazioni individuate e dei costi relativi hanno comportato un’assegnazione delle priorità tra i requisiti individuati e la pianificazione di futuri sviluppi della piattaforma.

5 Analisi degli scostamenti su attività e risultati

Per quanto concerne la LA 2.14 non si devono segnalare scostamenti rispetto alle attività e ai risultati attesi.

6 Sintesi delle attività svolte

L'attività ha riguardato la descrizione formale dei requisiti funzionali e architeturali (ma non prestazionali) per l'implementazione della nuova piattaforma denominata "CI-RES" (Critical Infrastructure - RESilience), che costituisce una re-ingegnerizzazione della piattaforma CIPCAST sviluppata nel triennio precedente. La necessità di gestire diverse applicazioni, sviluppate da ENEA nell'ambito del precedente e corrente piano triennale, ha determinato la specifica di un'architettura modulare, flessibile con strumenti idonei all'integrazione delle funzionalità della piattaforma e dei dati da essa prodotti in applicazioni, piattaforme esterne come ad esempio la Smart City Platform.

7 Dettaglio delle attività svolte

L'obiettivo primario della piattaforma CI-RES è costituire lo strumento informatico di riferimento per la valutazione della resilienza delle infrastrutture critiche. Questa attività si inserisce in un piano generale di organizzazione e gestione delle varie attività di sviluppo del progetto 1.7; in questo senso, la presente attività, insieme alla attività portata avanti nel WP1 LA16 (realizzazione di un cluster Kubernetes in produzione, si veda report RdS_PTR 22-24_PR 1.7_LA1.16_003) è di vitale importanza per la gestione dei servizi, applicazioni, algoritmi definiti nell'ambito del presente piano triennale.

Come anticipato nella sezione 4 si è optato per la realizzazione di un'architettura **modulare**. Le applicazioni, servizi sviluppati nei vari moduli, che possono essere moduli di frontend, backend, acquisizione dati come meglio descritto in seguito, devono essere gestiti da uno o più componenti isolati ed autonomi seguendo un pattern architetturale denominato a **micro-servizi**. Questo pattern architetturale, molto diffuso recentemente, nell'ambito della presente attività è stato opportunamente esteso per venire incontro alle esigenze progettuali specifiche di CI-RES. Infatti, per non dovere gestire un numero molto elevato di micro-servizi si è deciso di proporre **meso-servizi**, cercando di scomporre applicazioni monolitiche nel numero minimale di componenti possibili.

Per la gestione di componenti opportunamente isolate ed autonome si è deciso di optare per la soluzione fornita dalla tecnologia **DOCKER**. Tramite Docker è possibile eseguire un servizio o applicazione specifica in elementi (container) gestiti tramite una virtualizzazione dell'ambiente di esecuzione del servizio stesso, dal sistema operativo fino alle librerie di supporto necessarie. Questo garantisce la possibilità di gestire su una stessa macchina diversi container ognuno gestibile in maniera autonoma per quanto l'aggiornamento del codice dell'applicazione stessa e delle librerie di supporto senza incorrere in possibili problemi di incompatibilità con altri componenti, servizi. Per l'architettura a meso-servizi di CI-RES e per i relativi container Docker sono stati individuati i seguenti requisiti:

- 1) Ogni docker che fornirà una procedura invocabile/configurabile dall'esterno dovrà essere dotato di un'interfaccia Web di tipo REST API.
- 2) I docker dovranno essere eseguibili, all'occorrenza, sulle diverse workstation disponibili in laboratorio.
- 3) I docker della piattaforma CI-RES dovranno essere "migrabili" nella futura piattaforma Kubernetes¹, che è uno degli obiettivi di laboratorio. La gestione sul cluster Kubernetes di questi docker consentirà un'erogazione più affidabile di questi servizi, consentendo piani di upgrade hardware e software senza interruzioni di servizio.
- 4) Per tali ragioni il server in produzione dovrà ospitare un **repository di docker** (docker Hub) che dovrà essere accessibile dal team.

Requisiti della piattaforma CI-RES

La piattaforma CI-RES può essere suddivisa logicamente in tre blocchi: A) CI-RES Front End, B) CI-RES Core e C) CI-RES Back End.

- A) Il **CI-RES Core** è costituito da tre componenti principali:
- una collezione di **database consolidati** (già esistenti nella piattaforma CIPCAST e nei vari server/workstation ENEA), dove sono presenti i dati (geospaziali e non), che possono essere utilizzati per la creazione di mappe e per attività di analisi dal team e accessibili dal backend. In particolare, un database PostGIS in cui migrare i dati di CIPCAST.

¹ <https://kubernetes.io/it/>

- Un'istanza di GeoNode che andrà a costituire anche il principale portale per il team come strumento di visualizzazione e condivisione di mappe. Tale istanza dovrà essere opportunamente configurata per rispondere alle esigenze specifiche del team di lavoro e dovrà contenere il suo GeoServer autonomo e il suo PostGIS autonomo.
 - Un GeoServer aggiuntivo per la gestione dei dati georeferenziati accessibile anche dal frontend e dal backend. Tipici esempi: i layer contenenti dati atmosferici previsionali, mappe di “ground-motion” da dati interferometrici satellitari, dati dalla piattaforma ENEA-PELL, mappe di rischio dinamico etc.
- B) Il blocco **CI-RES Backend** sarà costituito dalle procedure, (automatizzate e non), dai tool e dalle metodologie che costituiscono il livello logico di business, ovvero gli algoritmi e la loro implementazione informatica.
- C) Il blocco **CI-RES Frontend** ospiterà gli strumenti messi a disposizione dalla piattaforma per l'accesso, la visualizzazione di mappe e dati e per rispondere ai requisiti di particolari utenti. Come vedremo in dettaglio nel seguito, il Frontend sarà formato da: i) portali web “verticali” che consentiranno la visualizzazione di mappe e dati specifici; ii) da un determinato numero di web services end-point che risponderanno alle varie esigenze della piattaforma.

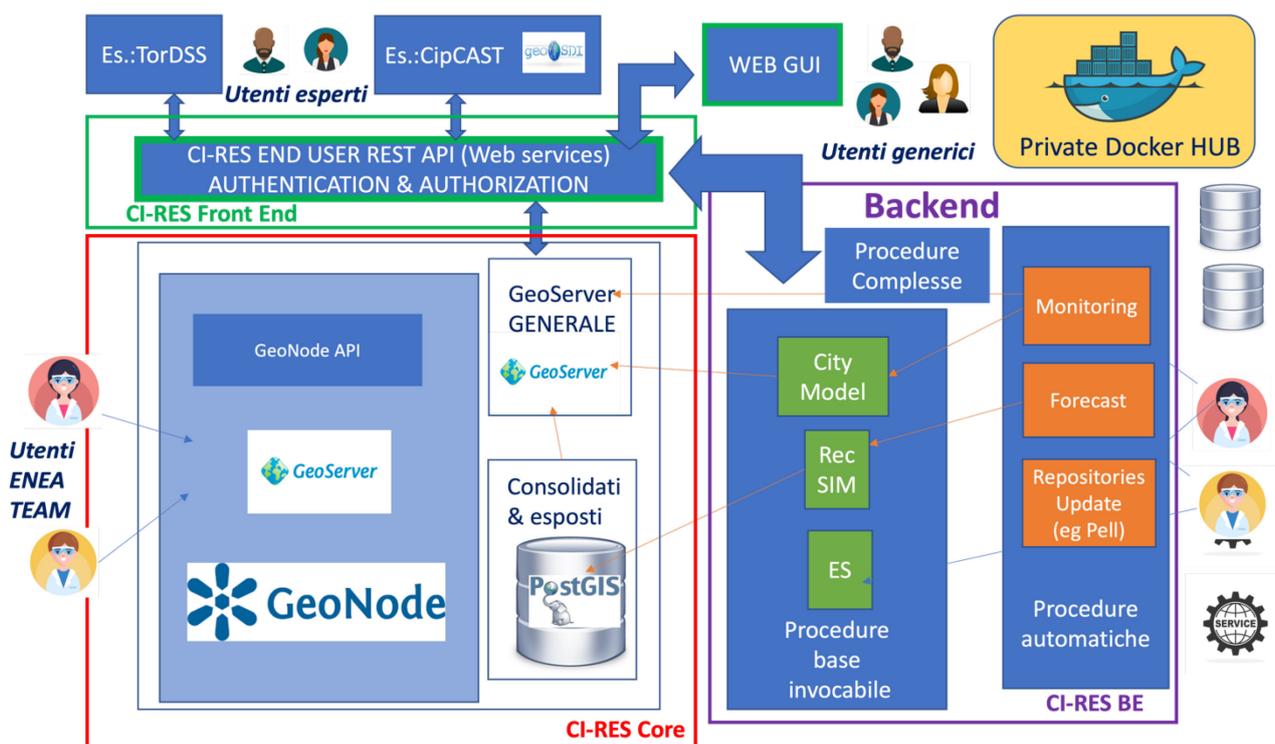


Figura 1 - Architettura a blocchi di CI-RES

La Figura 1 mostra uno schema con i vari componenti, divisi nei tre macro-blocchi descritti precedentemente, dell'architettura complessiva della piattaforma CI-RES. Si noti la presenza di un applicativo specifico (SCP WEB-I) che rappresenta l'interfaccia per l'interazione con la Smart City Platform (SCP).

- 5) L'architettura risponde all'esigenza di evoluzione dinamica dei codici (APIC Docker Hub) e dei dati (DB consolidati) utilizzati per la modellizzazione, analisi e protezione delle infrastrutture critiche ed in modo particolare del sistema di sistemi da esse risultante. Uno degli obiettivi principali della re-ingegnerizzazione è, infatti, la definizione di una metodologia per lo sviluppo e l'integrazione di applicativi utilizzando le diverse componenti previste dall'architettura.

Per questo motivo, come anticipato, per la piattaforma CI-RES si dovranno creare due ambienti distinti:

1. **ambiente di sviluppo**
2. **ambiente di produzione.**

L'ambiente di sviluppo sarà utilizzato per l'implementazione delle funzionalità di base ed eventualmente, in futuro, di funzionalità aggiuntive; cioè per creare i nuovi docker da integrare sulla piattaforma in produzione. Quindi, anche la modalità operativa di creazione della piattaforma prevede la "dockerizzazione" delle varie componenti per il loro deployment nell'ambiente di produzione.

I due ambienti saranno realizzati su due server Dell R650 con sistema operativo CentOS Stream 9. Le due macchine saranno integrate con un server di backup dati (da definire). La macchina in produzione avrà sia un IP pubblico che uno privato della LAN del CR ENEA Casaccia.

La Figura 2 mostra il tipico diagramma di flusso della metodologia di sviluppo dei nuovi applicativi che si intende realizzare nell'ambito delle attività future sulla piattaforma CI-RES.

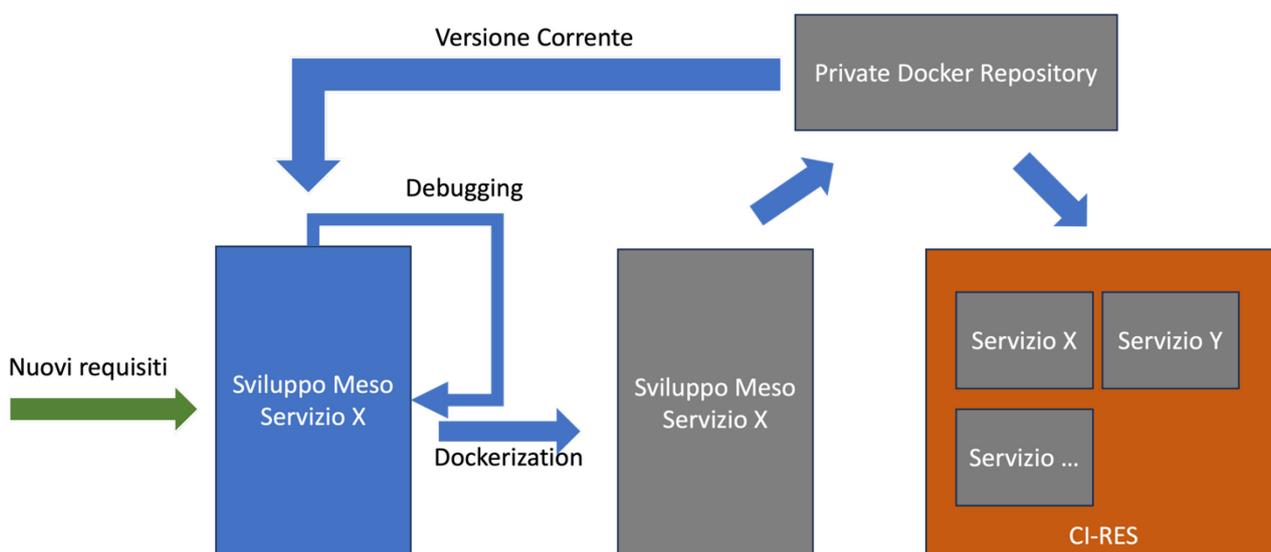


Figura 2 - Schema di flusso per lo sviluppo delle procedure e del loro update sulla piattaforma CI-RES

In particolare, la figura mostra la centralità nell'architettura individuate dell'Private Docker Repository (Private Docker HUB in Figura 1), da utilizzare per la gestione delle varie immagini Docker generate per la realizzazione dei diversi servizi di CI-RES sviluppate dall'ENEA. Come descritto precedentemente l'architettura proposta non segue un pattern architetturale a micro-servizi nel senso stretto del termine. In CI-RES, un'immagine Docker potrebbe realizzare un servizio accorpando diverse funzionalità che potrebbero essere implementate come micro-servizi (come, ad esempio, l'acquisizione di dati sugli eventi sismici in tempo reale da INGV ed il calcolo delle relative *shakemap*). Per questo motivo, come detto, nell'ambito di questa re-ingegnerizzazione i servizi docker sono denominati **meso-servizi**. In fasi di sviluppo successive i meso-servizi più complessi saranno parcellizzati dockerizzando le loro componenti. La scelta attuale consente la continuità dei servizi esistenti e la loro rapida portabilità sulla nuova piattaforma.

Requisiti CI-RES Core

Come detto nell'introduzione del presente documento, i **componenti principali del CI-RES Core** sono:

- il portale GeoNode con il proprio GeoServer (e relativi componenti: e.g., PostgreSQL-PostGIS, etc.);
- Le banche dati (database) contenenti dati consolidati ed esposti;

- un GeoServer (di seguito denominato GeoServer CI-RES) per la pubblicazione dei layer verso utenti esterni e per la gestione di layer “dinamici” come, ad esempio, i layer previsionali meteo. Come verrà illustrato in seguito, un utente di GeoNode (quindi un membro del team APIC) potrà decidere se “consolidare” (ossia memorizzare ed esportare) un layer dinamico in GeoNode per la creazione di mappe, per eseguire opportune analisi (come ad esempio la valutazione di impatto di un possibile scenario di crisi possibilmente generato da determinate previsioni meteo).
- **GeoNetwork** per la gestione dei metadati nel modello di integrazione (da definire)

La Tabella 1 riporta i principali **requisiti per l’architettura del core**. Il simbolo S nell’ultima colonna indica un requisito desiderato opzionale, mentre M indica un requisito necessario:

Requisito	Descrizione	Priorità
<i>CI-RES Core Req 0</i>	Installazione GeoNode nell’ambiente di sviluppo	M
<i>CI-RES Core Req 1</i>	Creazione Immagine docker della versione di GeoNode opportunamente customizzata per rispondere ai requisiti individuati nel presente documento (di seguito GeoNode CI-RES)	M
<i>CI-RES Core Req 2</i>	Deployment di GeoNode CI-RES nel server di produzione: ovvero pubblicazione dell’immagine di GeoNode nel repository docker in produzione e esecuzione di una istanza come servizio del sistema operativo che parte al bootstrap.	M
<i>CI-RES Core Req 3</i>	Creazioni di opportune procedure di backup dei dati gestiti da GeoNode. Le procedure di backup faranno uso di risorse di memorizzazione messe a disposizione dal team APIC su dispositivi appartenenti alla LAN.	M
<i>CI-RES Core Req 4</i>	Creazione di una guida sintetica per facilitare le procedure di creazione immagine docker e deployment su server di produzione.	M
<i>CI-RES Core Req 5</i>	I vari workflow operativi e le procedure implementate nella piattaforma CI-RES utilizzano un database PostgreSQL con estensione PostGIS per la memorizzazione di dati di input/output. Tale requisito richiede l’installazione di un cluster PostgreSQL con estensione PostGIS nel server di sviluppo	M
<i>CI-RES Core Req 6</i>	Installazione di un cluster PostgreSQL con estensione PostGIS nel server di produzione [in docker con dati persistenti]	M
<i>CI-RES Core Req 7</i>	Configurazione del cluster PostgreSQL-PostGIS CI-RES con il requisito di High Availability	S
<i>CI-RES Core Req 8</i>	Installazione del GeoServer CI-RES (che gestirà tutti i layer della piattaforma CI-RES) nell’ambiente di sviluppo. Tale istanza di GeoServer non deve essere confusa con il GeoServer proprio di GeoNode.	M
<i>CI-RES Core Req 9</i>	Deployment dockerizzata del GeoServer CI-RES (che gestirà tutti i layer della piattaforma CI-RES) nell’ambiente di produzione	M
<i>CI-RES Core Req 10</i>	Importazione del DB di CIPCAST sul DB del cluster PostGIS CI-RES	S

Tabella 1 - Requisiti CI-RES Core

Alcune delle descritte applicazioni (ad esempio, la customizzazione di GeoNode menzionata nel requisito *CI-RES Core Req 1*) necessitano dell’**implementazione di nuove applicazioni Django** (estensioni) che dovranno coesistere con le applicazioni di default di GeoNode. Ognuna di queste applicazioni Django aggiuntive prevede lo sviluppo di **specifiche interfacce utente** per:

- a) l’acquisizione dei dati di input forniti dall’utente;
- b) la visualizzazione di eventuali output, per il controllo della visualizzazione dei layer informativi previsti dall’applicazione stessa.

Ovviamente queste applicazioni Django aggiuntive avranno dei controlli per consentire all’utente di rendere permanentemente fruibili su GeoNode i layer che l’utente vuole gestire tramite GeoNode (esempio: un layer di previsione meteo è di particolare interesse in quanto potrebbe comportare scenari critici, l’utente deve avere la possibilità di consolidare questo layer in GeoNode per l’utilizzo di future analisi, etc.).

Una applicazione considerata chiave per la nuova piattaforma sarà denominata **GeoNode Earthquake Simulator**, dovrà essere sviluppata in GeoNode e consentirà:

- la visualizzazione di mappe esterne a GeoNode (come, ad esempio, i dati sismici)
- la configurazione e l'esecuzione del componente (di back-end) Earthquake Simulator

L'Earthquake Simulator, componente dockerizzata di CI-RES BackEND, prende in input le caratteristiche fondamentali di un evento sismico:

- Epicentro
- Magnitudo
- Profondità
- Mappa della V_{S30}
- Caratteristiche della faglia sismogenetica (opzionale)
- Microzonazione sismica (opzionale)

inoltre, il simulatore prende in input:

- Leggi di attenuazione da utilizzare per la generazione delle shakemap (mappe di scuotimento)
- Dati topologici e caratteristici delle infrastrutture critiche.

Come output il simulatore genera una shakemap e gli strati informativi geospaziali nei quali sono mappati e classificati (secondo differenti livelli di danneggiamento atteso) gli elementi (edifici o infrastrutture critiche, come ad esempio, acquedotti) impattati dall'evento sismico.

L'invocazione dell'Earthquake Simulator avverrà tramite le opportune interfacce http REST (REpresentational State Transfer) API definite in dettaglio nei requisiti della componente di Back End come mostrato in Figura 3.

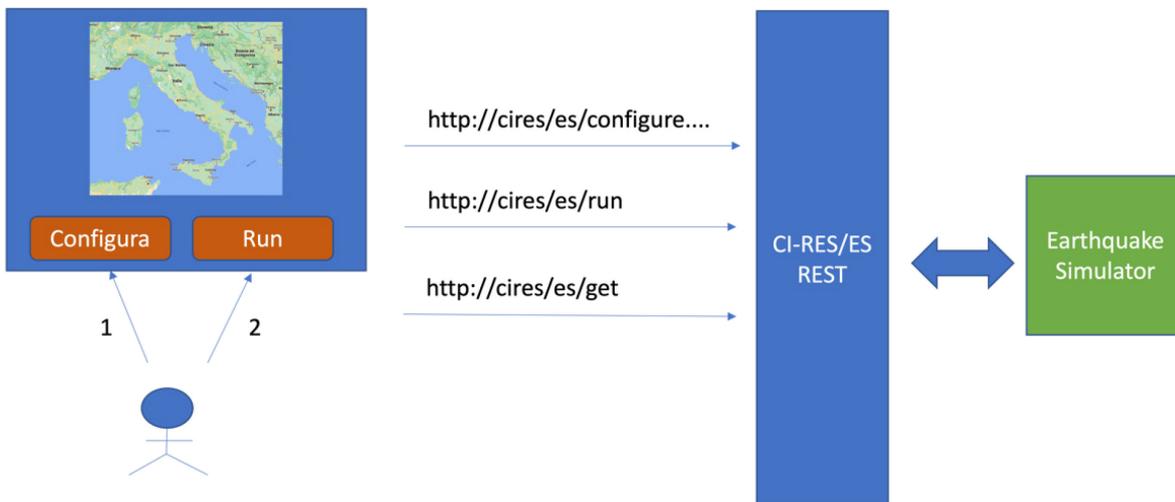


Figura 3 - Invocazione simulatore sismico

La Figura 3 mostra alcuni elementi dell'interfaccia grafica da realizzare per l'invocazione del simulatore di scenari sismici. Questo è un esempio di invocazione di una procedura standalone ed invocabile. Come si evince dalla Figura 3, l'interfaccia del simulatore è costituita da una serie API di tipo REST. Questo, come detto, è un pattern architetturale che si vorrebbe proporre come standard per le procedure di back end di tipo invocabile. Nella seguente Tabella 2 sono riportati i requisiti per l'applicazione Earthquake Simulator (ES).

Requisito	Descrizione	Priorità
<i>CI-RES Core Req 11</i>	Realizzazione nel GeoNode di sviluppo dell'applicazione Earthquake Simulator (ES). Tale applicazione deve essere visualizzata tramite opportuni elementi grafici (link, button) dalla landing page di GeoNode	M
<i>CI-RES Core Req 12</i>	L'applicazione ES deve essere utilizzabile solo dagli utenti abilitati	M
<i>CI-RES Core Req 13</i>	L'applicazione deve visualizzare una mappa di base (e.g. OpenstreetMap, Google Satellite) per consentire il posizionamento dell'epicentro dell'evento sismico. Tali coordinate saranno riportate su opportuni campi dell'interfaccia grafica	M
<i>CI-RES Core Req 14</i>	L'applicazione deve prevedere un opportuno form di input (menu a tendina) per l'inserimento dei parametri della simulazione (e.g. magnitudo dell'evento sismico, legge di attenuazione, etc.)	M
<i>CI-RES Core Req 15</i>	L'applicazione deve prevedere un opportuno form di input (menu a tendina) per la selezione dell'infrastruttura di interesse i cui dati saranno utilizzati nella simulazione	M
<i>CI-RES Core Req 15 BIS</i>	In base alla scelta utente relativa alle coordinate dell'epicentro dell'evento sismico (CI-RES Core Req 13), l'applicazione carica i layer degli elementi esposti (edifici, scuole, reti infrastrutture) presenti nel catalogo e che insistono in un'area nell'intorno dell'epicentro	M
<i>CI-RES Core Req 16</i>	L'applicazione deve prevedere un button per l'avvio della simulazione	M
<i>CI-RES Core Req 17</i>	Al termine della simulazione dovrà essere visibile (oppure cliccabile) un button per la visualizzazione del (o dei) layer (tipicamente in formato vettoriale), con la mappatura degli elementi dell'infrastruttura scelta impattati dall'evento sismico, nonché del layer relativo alla shakemap calcolata dal simulatore (tipicamente in formato raster)	M
<i>CI-RES Core Req 18</i>	Al termine della simulazione dovrà essere visibile (oppure cliccabile) un button per il caricamento dei layer ottenuti come risultato della simulazione su GeoNode. L'utente, in tal caso potrà compilare mediante form di input i metadati da associare alla mappa, oltre che indicare i permessi di utilizzo e visualizzazione	M

Tabella 2 - Requisiti applicazione Earthquake Simulator (ES)

8 Contributo delle eventuali consulenze alle attività sopra descritte

Nessuna consulenza è prevista in questa LA.

9 Pubblicazioni scientifiche

Nessuna pubblicazione scientifica.

10 Eventi di disseminazione

Nessun evento di disseminazione.