MISSION INNOVATION ITALY

MINISTERO DELL'AMBIENTE
E DELLA SICUREZZA ENERGETICA

D1.9 - Descrizione del DB ENEA per i materiali per l'energia a servizio della piattaforma IEMAP

Sergio Ferlito, Massimo Celino

ENEA
AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

Report MI21-24/132

MISSION INNOVATION ITALY

MINISTERO DELL'AMBIENTE
E DELLA SICUREZZA ENERGETICA

D1.9 - Descrizione del DB ENEA per i materiali per l'energia a servizio della piattaforma IEMAP

Sergio Ferlito (ENEA), Massimo Celino (ENEA)

Dicembre 2024

Report MISSION INNOVATION

Ministero dell'Ambiente e della Sicurezza Energetica - ENEA
Mission Innovation 2021-2024 - III annualità

Progetto: Piattaforma accelerata per i Materiali per l'Energia
Work package: MAP: Italian Energy Materials Acceleration Platform
Linea di attività 1.6: Utilizzo DB ENEA
Responsabile del Progetto: Massimo Celino (ENEA)
Responsabile della LA: Marialuisa Mongelli (ENEA)

Indice

Indice delle figure

Indice delle tabelle

1 Introduzione

1.1 Contesto del progetto e ruolo del DB ENEA

l progetto IEMAP (Piattaforma Italiana Accelerata per i Materiali per l'Energia), inserito nell'ambito dell'iniziativa Mission Innovation, mira a sviluppare un'infrastruttura digitale integrata per supportare la comunità scientifica nazionale nella gestione, elaborazione e valorizzazione dei dati relativi ai materiali avanzati per l'energia. In questo ecosistema, il database sviluppato e gestito da ENEA costituisce il nucleo centrale e l'abilitatore fondamentale di tutte le funzionalità della piattaforma.
Il database non è concepito come un semplice repository passivo, ma come un sistema attivo e strutturato, progettato per raccogliere in un unico bacino dati eterogenei, provenienti sia da simulazioni computazionali che da processi sperimentali condotti dai diversi partner di progetto. La sua funzione strategica è quella di centralizzare l'informazione, garantendone l'accesso regolamentato e la persistenza nel tempo, al fine di promuovere la condivisione, la riproducibilità delle ricerche e l'applicazione di tecniche avanzate di analisi, inclusi modelli di Intelligenza Artificiale. L'infrastruttura di database ENEA, pertanto, è l'elemento portante su cui poggiano il front-end web, i servizi REST API e il modulo Python ***iemap-mi***, che insieme compongono l'interfaccia computazionale della **piattaforma IEMAP** (per ulteriori dettagli si veda il deliverable LA 1.6_DA1.03 Piattaforma IEMAP)

1.2 Obiettivi del deliverable

Il presente documento ha l'obiettivo di fornire una descrizione completa e dettagliata del database ENEA a servizio della piattaforma IEMAP. Verranno illustrate le scelte architetturali, le modalità operative e i contenuti del database, con un'attenzione specifica ai processi implementati per garantire la qualità, la coerenza e la sicurezza dei dati raccolti.
In particolare, il deliverable si propone di:

- **Illustrare il flusso condiviso di raccolta dei dati**, descrivendo l'utilizzo di template standardizzati e le procedure di validazione adottate per l'inserimento di metadati e file.
- **Dettagliare le modalità operative di gestione del dato,** incluse le tecnologie impiegate (MongoDB Enterpris & Ceph), le politiche di conservazione, backup e versionamento.
- **Descrivere in modo approfondito il contenuto informativo del datab**ase, specificando le tipologie di dati archiviati e la loro organizzazione logica in collezioni distinte.
- **Valutare il livello di aderenza del database ai principi FAIR** (Findable, Accessible, Interoperable, Reusable), analizzando i punti di forza e identificando le aree di potenziale miglioramento per massimizzare il valore del patrimonio informativo raccolto.

Questo documento è inteso come una guida tecnica di riferimento per tutti gli stakeholder del progetto IEMAP, dai ricercatori che contribuiscono con i propri dati agli sviluppatori che interagiscono con la piattaforma tramite le sue interfacce programmatiche.

2 Processo condiviso di raccolta dati

La coerenza e l'affidabilità del database ENEA dipendono da un processo di raccolta dati meticolosamente strutturato e condiviso tra tutti i partner del progetto IEMAP. Per garantire l'omogeneità delle informazioni, massimizzare la loro interoperabilità e semplificare il conferimento dei dati da parte di ricercatori con diversi background tecnici, è stato definito un flusso operativo standard. Questo flusso governa in modo rigoroso sia l'inserimento dei metadati descrittivi sia il caricamento dei file associati a ciascun esperimento o simulazione. L'approccio centralizzato e semi-guidato è stato progettato per ridurre l'ambiguità semantica, prevenire l'inserimento di dati incompleti e facilitare le successive, e cruciali, fasi di interrogazione, analisi e valorizzazione del patrimonio informativo raccolto.

Descrizione del flusso di inserimento metadati e file

L'inserimento di un nuovo dataset sulla piattaforma IEMAP è concepito come un processo sequenziale e atomico, articolato in due fasi distinte che separano nettamente la descrizione contestuale dei dati (i metadati) dai dati grezzi o elaborati (i file).
Tale separazione è una scelta architetturale deliberata, finalizzata a garantire che nessun dato possa esistere nel repository senza un adeguato contesto descrittivo.

Fase 1: Inserimento e Validazione dei Metadati
Il primo passo fondamentale consiste nella creazione di un "documento" di progetto all'interno del database NoSQL MongoDB. Questo documento funge da contenitore logico per tutte le informazioni relative a una specifica attività sperimentale o computazionale. L'intero processo è guidato dall'interfaccia web e richiede che l'utente sia autenticato come partner di progetto.
Il flusso operativo per l'utente è il seguente:

1. **Download del Template**: Dalla sezione dedicata ai progetti, l'utente-partner scarica il file **metadata_template.xlsx**. Questo file Excel è il principale strumento di standardizzazione per la raccolta dei metadati.
2. **Compilazione Offline**: Il partner compila il template in locale, seguendo la struttura a fogli di lavoro e le regole di validazione integrate, descritte in dettaglio nel paragrafo successivo. Ogni file Excel compilato corrisponde a un singolo progetto, a cui è associato uno e un solo processo (sperimentale o computazionale).
3. **Upload e Pre-visualizzazione**: Una volta compilato, il file Excel viene caricato sulla piattaforma tramite l'apposito pulsante "Browse metadata file". Il front-end dell'applicazione, eseguito nel browser dell'utente, legge il file, ne estrae i dati e li converte in formato JSON (JavaScript Object Notation). A questo punto, l'interfaccia mostra un'anteprima del JSON generato, permettendo all'utente di verificare la corretta interpretazione dei dati prima dell'invio definitivo al server.
4. **Creazione del Progetto**: Con un clic sul pulsante "Create Project", il payload JSON validato viene inviato al back-end della piattaforma. Qui, dopo un'ulteriore validazione lato server, viene creato il documento corrispondente nella collezione MongoDB, a cui viene assegnato un identificatore univoco (iemap_id, generato automaticamente dal back-end).

Fase 2: Associazione e Archiviazione dei File

Completata con successo la creazione del record di metadati, l'utente viene automaticamente reindirizzato alla pagina di dettaglio del progetto appena inserito. Questa pagina funge da dashboard per il progetto e costituisce il punto di accesso per la seconda fase del processo: l'associazione dei file.

1. **Upload dei File Associati**: Attraverso il pulsante "ADD FILES", l'utente può caricare uno o più file (es. report in PDF, immagini, dati di output, file .cif) da associare al progetto. Per garantire la responsività del server e gestire anche file di grandi dimensioni (fino a 1 GB), l'upload avviene in modo incrementale, attraverso uno stream di dati a blocchi con dimensione massima di 1 MB.
2. **Calcolo dell'Hash e Archiviazione**: Una volta completato il caricamento di un file, il back-end esegue due operazioni cruciali in sequenza. Per prima cosa, calcola un **hash crittografico** (es. SHA-256) del contenuto del file. Questo hash è una stringa alfanumerica univoca che funge da impronta digitale: garantisce l'integrità del dato (qualsiasi modifica al file produrrebbe un hash diverso) e permette di identificare facilmente eventuali duplicati. Successivamente, il file viene salvato fisicamente su un **filesystem distribuito Ceph**, un sistema di storage ad alta affidabilità e scalabilità. Il file viene rinominato utilizzando il suo hash, secondo la convenzione *<hash_file>.<estensione_file>.*
3. **Collegamento nel Database**: Infine, le informazioni relative al file caricato (l'hash, il nome originale, l'estensione, la dimensione, la data di creazione) vengono aggiunte a un array denominato files all'interno del documento MongoDB del progetto. In questo modo, il database centrale non contiene i file veri e propri, ma solo i metadati e i puntatori univoci (gli hash) ai file fisici archiviati su Ceph.

Questo flusso a due stadi assicura che ogni dato sia sempre rintracciabile, contestualizzato e gestito in modo sicuro e scalabile, separando la logica descrittiva (MongoDB) dalla logica di archiviazione massiva (Ceph).

Utilizzo di template e controlli di validazione

Il cuore del processo di standardizzazione dei dati è il **template Excel metadata_template.xlsx**. Questo strumento è stato progettato non come un semplice foglio di calcolo, ma come un modulo interattivo semi-guidato che impiega regole di validazione interne, espandibili e non rigidi, per minimizzare gli errori di inserimento e garantire la coerenza terminologica e formale tra tutti i dati conferiti dai partner. Tale scelta pur consentendo una validazione e parziale inserimento “guidato” è stato anche man mano ampliato su segnalazione dei partner permettendo anche un certo grado di adattamento che ben si sposa con le caratteristiche di schema flessibile offerte da MongoDB.

Struttura del Template

Il file è organizzato in sei fogli di lavoro, di cui cinque destinati all'inserimento dati e uno a scopo informativo:

- **how_to**: Contiene le istruzioni per la compilazione e, soprattutto, le liste di valori predefiniti (ma espandibile) utilizzati per la validazione dei dati negli altri fogli.
- **Project**: Per inserire il nome del progetto/attività e la relativa etichetta. Le scelte sono guidate da un menù a discesa che include opzioni predefinite come "Materials for Batteries" (MB), "Materials for Electrolyzers" (ME), "Materials for Perovskite" (MPV), ecc..

- **Process**: Per descrivere la simulazione o l'esperimento. I campi includono method (es. DFT), agent.name (lo strumento software o hardware), agent.version e isExperiment (un booleano per distinguere tra attività computazionali e sperimentali).
- **Materials**: Per specificare la formula chimica del materiale oggetto dello studio. Questo campo non ha regole di validazione predefinite per consentire la massima flessibilità.
- **Parameters** e **Properties**: Questi fogli permettono di inserire i parametri di input e le proprietà misurate o calcolate. Ogni voce è strutturata come una tripla di "nome", "valore" e "unità" (es. "temperatura", "20", "gradi centigradi"). Anche qui, i nomi dei parametri e delle proprietà sono spesso selezionabili da liste predefinite per mantenere la coerenza.

Meccanismi di Validazione e Flessibilità

Molte celle del template utilizzano la funzionalità di **"Convalida dati" di Excel**, che limita la scelta a un set di valori pre-approvati e definiti nel foglio how_to. Questo approccio previene l'uso di sinonimi, abbreviazioni non standard o errori di battitura, che comprometterebbero la ricercabilità dei dati.
La piattaforma, tuttavia, è progettata per essere flessibile. Qualora un partner necessiti di inserire un nuovo valore (ad esempio, un nuovo metodo sperimentale) non presente negli elenchi, può aggiungerlo manualmente alla lista corrispondente nel foglio how_to e aggiornare l'intervallo di validazione della cella pertinente. La documentazione della piattaforma include una guida passo-passo per eseguire questa operazione, garantendo un'espansione controllata del vocabolario condiviso.
Oltre alla validazione lato client offerta dal file Excel, la piattaforma IEMAP implementa un **doppio livello di controllo** per massimizzare la robustezza:

1. **Validazione sul Front-end**: Prima dell'invio al server, il codice JavaScript in esecuzione nel browser controlla la struttura e la completezza del JSON estratto dal file.
2. **Validazione sul Back-end**: Prima del salvataggio definitivo nel database, i dati in formato JSON vengono ulteriormente validati da **modelli Pydantic**. Pydantic è una libreria Python che permette di definire schemi di dati con tipi e vincoli specifici. Se i dati ricevuti non sono conformi allo schema (es. un campo obbligatorio è mancante, o un valore ha un tipo errato), la richiesta viene rigettata e viene restituito un errore informativo. Questo garantisce l'integrità referenziale e la coerenza strutturale di ogni singolo documento nel database.

Ruoli e responsabilità dei partner nel processo

Il successo del popolamento del database si basa su una chiara e imprescindibile suddivisione dei ruoli e delle responsabilità. L'accesso alla piattaforma IEMAP è differenziato: la consultazione dei progetti e dei dati è pubblica e non richiede autenticazione, in linea con i principi di apertura della ricerca. Al contrario, la capacità di contribuire al patrimonio informativo, ovvero di creare nuovi progetti e caricare dati, è riservata esclusivamente ai **partner del progetto**, che devono disporre di un account utente registrato e validato sulla piattaforma.

Processo di Abilitazione del Partner

Per diventare un contributore, un utente deve seguire una procedura di registrazione sicura:

1. **Registrazione**: L'utente si registra fornendo un indirizzo e-mail valido, la propria affiliazione (selezionata da un elenco di partner del progetto) e una password.
2. **Validazione via E-mail**: Il sistema crea un account temporaneo e invia un'e-mail di verifica all'indirizzo fornito. L'utente deve fare clic sul link di attivazione contenuto nell'e-mail per validare il proprio account.
3. **Scadenza dell'Account Temporaneo**: Per ragioni di sicurezza, l'account non validato viene automaticamente eliminato dal database dopo 24 ore, grazie a un indice TTL (Time-To-Live) impostato in MongoDB. L'utente dovrà quindi ripetere la procedura se non completa la validazione in tempo.

Responsabilità del Partner Contributore

Una volta abilitato, il partner assume la piena responsabilità della **qualità, accuratezza e completezza** dei dati che conferisce. Questo include:

- La **corretta compilazione del template Excel**, assicurandosi che i metadati descrivano in modo esaustivo e non ambiguo l'esperimento o la simulazione.
- Il **caricamento di tutti i file rilevanti** associati all'attività, verificando che siano nei formati supportati dalla piattaforma (es. PDF, DOCX, XLSX, PNG, JPG, CIF, TXT, CSV) e che non superino il limite di dimensione di 1 GB per file.
- La **verifica della coerenza** tra le informazioni inserite nei metadati (es. la formula di un materiale) e il contenuto effettivo dei file caricati.

Il team di gestione della piattaforma ENEA, d'altra parte, ha la responsabilità di mantenere l'infrastruttura tecnologica, assicurare il corretto e sicuro funzionamento del flusso di inserimento, gestire i backup e fornire supporto tecnico tempestivo ai partner in caso di difficoltà o segnalazione di anomalie. **La qualità complessiva del database IEMAP è, in ultima analisi, un risultato sinergico della robustezza della piattaforma tecnologica e della diligenza scientifica dei suoi contributori.**

3 Modalità operative di raccolta, pulizia e conservazione

La robustezza e l'affidabilità della piattaforma IEMAP si fondano su un insieme di modalità operative rigorose che governano l'intero ciclo di vita del dato: dalla sua acquisizione iniziale fino alla sua conservazione sicura a lungo termine. L'architettura implementata adotta un approccio modulare e stratificato, dove ogni fase del processo è gestita da componenti tecnologiche specifiche e procedure standardizzate. Questo capitolo descrive in dettaglio i meccanismi di raccolta, le procedure di pulizia e validazione, l'infrastruttura di conservazione e le politiche di sicurezza e backup, che insieme garantiscono l'integrità, la qualità e la perennità del patrimonio informativo del progetto.

3.1 Modalità di Raccolta Dati

La piattaforma IEMAP è stata progettata per essere flessibile e accessibile a diverse tipologie di utenti e casi d'uso. Per questo motivo, sono stati predisposti molteplici canali di ingestione dei dati, ciascuno ottimizzato per uno specifico scenario di interazione, dall'inserimento manuale da parte di un

ricercatore all'integrazione automatizzata in pipeline computazionali complesse. Tutti i canali convergono verso un unico set di servizi back-end, garantendo che, indipendentemente dal punto di ingresso, i dati vengano processati e validati in modo uniforme

3.2 Interfaccia Web (front-end)

L'interfaccia web, accessibile all'indirizzo **https://iemap.enea.it**, rappresenta il principale canale di interazione per gli utenti umani. È stata sviluppata come una Single Page Application (SPA) utilizzando il framework VueJS v3, una scelta tecnologica che garantisce un'esperienza utente fluida, reattiva e moderna. Questo canale è ottimizzato per la raccolta guidata di metadati e file. Il processo, come descritto nel capitolo precedente, si basa sull'upload del template Excel. Dal punto di vista operativo, questa interazione scatena una serie di eventi tecnici:

- **Conversione Client-Side**: Una volta caricato il file .xlsx, il codice JavaScript eseguito nel browser dell'utente (utilizzando librerie specifiche) ne effettua il parsing, estrae i valori dai fogli di lavoro e li converte in un oggetto JSON strutturato, pronto per essere inviato al server.
- **Comunicazione Asincrona**: La comunicazione tra il front-end e il back-end avviene tramite chiamate HTTP asincrone, gestite dalla libreria Axios. Questo permette all'interfaccia di rimanere reattiva mentre i dati vengono trasmessi e processati dal server, senza richiedere il ricaricamento completo della pagina.
- **Upload Incrementale dei File**: Per la gestione di file di grandi dimensioni (fino a 1 GB), l'upload non avviene in un unico blocco. Il file viene invece frammentato e inviato al server come uno stream di dati a blocchi (con dimensione massima di 1 MB). Questo approccio è fondamentale per la stabilità del sistema, in quanto previene il timeout delle richieste e riduce il consumo di memoria sul server, consentendo di gestire upload simultanei in modo efficiente.

3.3 Modulo Python

https://pypi.org/project/iemap-mi/
pip install iemap-mi

Riconoscendo la necessità di supportare l'automazione e l'integrazione della piattaforma in workflow scientifici programmatici, è stato sviluppato e pubblicato su PyPI (Python Package Index) il modulo open-source iemap-mi. Questo pacchetto software costituisce il canale di ingestione privilegiato per utenti avanzati, sviluppatori e data scientist. Esso astrae la complessità delle chiamate dirette alle API REST, fornendo un'interfaccia Pythonic ad alto livello.
Le sue caratteristiche operative principali sono:

- **Operazioni Asincrone**: Il modulo è costruito su httpx, una moderna libreria client HTTP che supporta nativamente le operazioni asincrone (async/await). Ciò lo rende estremamente performante e adatto a essere integrato in pipeline di dati ad alta intensità, dove la gestione non bloccante delle operazioni di rete è cruciale.

- **Validazione dati client-side & back-end**: Il sistema per l'inserimento dei dati come detto è duplice:
  - ✓ Via front-end: con validazione dei meta dati tramite libreria Zod
  - ✓ Via modulo iemap-mi: tale modulo incorpora modelli Pydantic (es. la classe IEMAPProject) che permettono di costruire e validare il payload dei metadati direttamente all'interno de modulo stesso

  Questo riduce il rischio di errori e fornisce un feedback immediato allo sviluppatore in caso di dati non conformi.

- **Handler Specializzati**: La libreria è organizzata in "handler" dedicati per specifici compiti, come ProjectHandler per la creazione e gestione dei progetti, AIHandler per l'interazione con i modelli predittivi, e StatHandler per il recupero di statistiche. Questa architettura modulare rende il codice pulito, leggibile e facilmente mantenibile.

3.4 Interfaccia REST API

A fondamento di entrambi i canali sopra descritti vi è l'interfaccia REST API, che rappresenta il vero e proprio punto di ingresso universale e machine-to-machine della piattaforma. Sviluppata in Python con il framework FastAPI, l'API espone un insieme di endpoint che permettono di eseguire tutte le operazioni CRUD (Create, Read, Update, Delete) sui dati. Dal punto di vista operativo, l'API è caratterizzata da:

- **Statelessness**: Ogni richiesta HTTP inviata a un endpoint API è atomica e indipendente. Il server non mantiene alcuna informazione di sessione tra una richiesta e l'altra. L'autenticazione, quando necessaria, viene gestita includendo un token JWT in ogni singola richiesta.
- **Standardizzazione**: L'API aderisce agli standard REST e utilizza i metodi HTTP convenzionali (es. POST per creare una risorsa, GET per leggerla). I dati vengono scambiati nel formato standard JSON.
- **Auto-documentazione**: Grazie a FastAPI, l'API genera automaticamente una documentazione interattiva tramite Swagger UI e ReDoc. Questa documentazione, accessibile pubblicamente, è uno strumento operativo fondamentale che permette agli sviluppatori di esplorare gli endpoint, comprendere i parametri richiesti e testare le chiamate direttamente dal browser.

3.5 Procedure di Pulizia e Validazione Dati

La qualità dei dati è un pilastro fondamentale del progetto IEMAP. Per garantire che le informazioni archiviate siano corrette, complete e coerenti, è stato implementato un robusto sistema di validazione multi-livello che agisce come un "quality gate", un cancello di qualità che i dati devono superare prima di essere resi persistenti nel database.

1. **Validazione a Livello Utente (Template Excel)**: La prima linea di difesa contro i dati "sporchi" è il template metadata_template.xlsx. L'uso di elenchi a discesa e regole di convalida dei dati integrate direttamente nel foglio di calcolo guida l'utente a inserire valori formalmente corretti e semanticamente coerenti con il vocabolario del progetto. Questo riduce drasticamente gli errori di battitura e l'uso di terminologie non standard.

2. **Validazione a Livello Client (Front-end)**: Prima che i dati lascino il browser dell'utente, un secondo livello di controllo viene eseguito dal codice JavaScript. Utilizzando librerie come @vee-validate/zod, il front-end verifica che il JSON generato dal file Excel sia strutturalmente corretto e che tutti i campi obbligatori siano presenti. Questo previene l'invio di richieste palesemente errate al server, migliorando l'efficienza e l'esperienza utente.
3. **Validazione a Livello Server (Back-end)**: Questo è il controllo più critico e inappellabile. Ogni dato che arriva agli endpoint della REST API viene sottoposto a una rigorosa validazione tramite i modelli Pydantic definiti nel codice del back-end. Pydantic confronta i dati in ingresso con uno schema predefinito, applicando controlli stringenti su:
    - **Tipi di dato**: Verifica che ogni campo abbia il tipo corretto (es. “isExperiment” deve essere un booleano, la proprietà “value” di un parametro deve essere un numero, ecc.).
    - **Campi Obbligatori**: Assicura che tutti i campi richiesti dallo schema siano presenti nel payload.
    - **Vincoli Aggiuntivi**: Può imporre ulteriori vincoli, come la lunghezza minima di una stringa o la conformità a un pattern (espressione regolare). Se anche una sola di queste regole non viene rispettata, Pydantic solleva un'eccezione, la richiesta viene rigettata e al client viene restituito un messaggio di errore HTTP 422 (Unprocessable Entity) contenente una descrizione dettagliata dell'errore. Questo meccanismo impedisce in modo categorico che dati malformati o incompleti possano contaminare il database.
4. **Validazione dei File**: Anche per i file esistono delle regole di validazione. Il sistema controlla l'estensione del file, accettando solo quelle presenti in una lista predefinita (csv, json, pdf, docx, cif, ecc.), e ne verifica la dimensione, rifiutando qualsiasi file che superi il limite di 1 GB.

3.6 Architettura di Conservazione Dati

L'architettura di storage è stata progettata per rispondere a due esigenze diverse ma complementari: la gestione flessibile e indicizzata di dati strutturati e semi-strutturati (i metadati) e l'archiviazione sicura e scalabile di file binari di grandi dimensioni. **Per questo motivo, è stata adottata una soluzione ibrida che combina un database NoSQL e un sistema di object storage distribuito.**

3.7 Conservazione dei Metadati: MongoDB Enterprise

Per i metadati è stata scelta l'edizione Enterprise di MongoDB, un database NoSQL orientato ai documenti, installato on-premise sull'infrastruttura di calcolo CRESCO di ENEA. La scelta di un database a documenti non è casuale: la sua flessibilità di schema si adatta perfettamente alla natura eterogenea dei dati scientifici, permettendo di archiviare esperimenti e simulazioni con set di parametri e proprietà molto diversi tra loro senza la rigidità di uno schema tabellare relazionale.

- **Schema del Documento**: Sebbene flessibile, la struttura dei documenti nella collezione iemap segue uno schema ben definito, che include sezioni nidificate come project (nome, etichetta, descrizione), process (metodo, agente), material (formula),

parameters e properties (array di oggetti nome/valore/unità), e provenance (informazioni sull'utente e timestamp createdAt/updatedAt per un versionamento di base).

- **Organizzazione delle Collezioni**: L'organizzazione dei dati è logica e granulare. I metadati principali dei progetti risiedono nel **database ai4mat, collezione iemap**. Dati specifici di un'attività, come quelli del processo di deposizione di perovskite, sono isolati in un database dedicato (perovskite) e archiviati in collezioni ottimizzate per il loro scopo: una **Time Series (TS) Collection** (sensors_readings) per le letture sequenziali dei sensori, e una collezione standard (source_events) per i dati statistici aggregati. Questa separazione garantisce performance ottimali per le query su ciascun tipo di dato.
- **Gestione Account Utente**: In un database separato, una collezione UserAuth gestisce le credenziali degli utenti. Su questa collezione è definito un **indice TTL (Time-To-Live)** sul campo is_verified. Questo meccanismo operativo fa sì che un account appena registrato, se non viene validato via e-mail entro 24 ore, venga automaticamente e in modo sicuro eliminato dal database.

3.8 Conservazione dei File: Ceph Distributed File System

Per l'archiviazione dei file associati ai progetti (documenti, immagini, dati grezzi) è stato scelto Ceph, un sistema di storage distribuito open-source noto per la sua elevata scalabilità, resilienza e performance. I file non vengono memorizzati all'interno di MongoDB per non appesantire il database e per sfruttare un sistema ottimizzato per la gestione di grandi oggetti binari.

- **Archiviazione Basata su Hash**: Il meccanismo di conservazione è intrinsecamente legato al **calcolo dell'hash crittografico**. Quando un file viene caricato, il suo contenuto viene utilizzato per generare un hash univoco. Il file viene quindi salvato su Ceph utilizzando questo hash come nome (<hash_file>.<estensione_file>). Questo approccio offre molteplici vantaggi operativi:
  - **Integrità del Dato**: L'hash funge da checksum. In qualsiasi momento è possibile ricalcolare l'hash del file archiviato e confrontarlo con quello memorizzato in MongoDB per verificare che il dato non sia stato corrotto.
  - **Immutabilità**: L'hash è legato al contenuto. Se un utente modifica anche un solo bit del file e lo ricarica, verrà generato un nuovo hash e il file verrà salvato come un nuovo oggetto, senza sovrascrivere il precedente. Questo crea un sistema di versionamento implicito e garantisce la non ripudiabilità del dato originale.
  - **De-duplicazione Automatica**: Se più utenti caricano lo stesso identico file, il sistema calcolerà lo stesso hash e, invece di creare una copia, semplicemente aggiungerà un nuovo riferimento in MongoDB allo stesso oggetto fisico già presente su Ceph, ottimizzando l'uso dello spazio di archiviazione.

3.9 Politiche di Backup e Sicurezza

La protezione del patrimonio informativo della piattaforma IEMAP è una priorità assoluta. A tal fine, è stata implementata una strategia di sicurezza multilivello che copre l'infrastruttura, le comunicazioni di rete, l'applicazione e i dati stessi.

3.10 Procedure di Backup e Disaster Recovery

La continuità operativa e la protezione contro la perdita di dati sono garantite da procedure di backup automatizzate e regolari. Il database MongoDB Enterprise è gestito tramite **MongoDB Ops Manager**, uno strumento che orchestra e automatizza le operazioni di backup. Vengono eseguiti snapshot periodici del database, che consentono di effettuare un ripristino **point-in-time** (a un preciso momento nel tempo) in caso di guasti hardware, corruzione logica dei dati o altri eventi imprevisti. I backup vengono archiviati in un luogo sicuro, separata dall'infrastruttura di produzione, per garantire la recuperabilità anche in caso di disastro a livello di data center.

3.11 Sicurezza dell'Infrastruttura e delle Comunicazioni

L'intera piattaforma è ospitata su un'infrastruttura virtualizzata **VMware** all'interno della rete ENEA.

- **Sicurezza Perimetrale**: L'accesso dall'esterno è filtrato e gestito da un server **NGINX**, che agisce come **reverse proxy**. Questo significa che i container Docker che eseguono i servizi applicativi non sono direttamente esposti su Internet, ma sono protetti da questo strato intermedio. NGINX è inoltre integrato con **Fail2ban**, un software che analizza i log e banna automaticamente gli indirizzi IP che mostrano comportamenti sospetti (es. tentativi di login falliti ripetuti), mitigando il rischio di attacchi brute-force.
- **Crittografia delle Comunicazioni**: Tutto il traffico tra il browser dell'utente e la piattaforma IEMAP è crittografato tramite il protocollo **HTTPS**. I certificati TLS/SSL necessari sono forniti da **Let's Encrypt** e il loro processo di richiesta e rinnovo periodico è completamente automatizzato tramite lo strumento **Certbot**, garantendo una protezione continua e senza interruzioni.

3.12 Sicurezza Applicativa: Autenticazione e Autorizzazione

L'accesso alle funzionalità di scrittura della piattaforma è strettamente controllato da un sistema di autenticazione e autorizzazione.

- **Autenticazione via JWT**: Il sistema si basa sullo standard **JSON Web Token (JWT)**. Al momento del login, il server genera un token firmato crittograficamente che contiene l'identità dell'utente. Questo token ha una scadenza limitata e deve essere incluso in ogni successiva richiesta a un endpoint protetto, all'interno dell'header HTTP Authorization: Bearer <token>.

- **Gestione Utenti con FastAPIUsers**: L'intero ciclo di vita dell'utente (registrazione, login, cambio password, validazione e-mail) è gestito dalla libreria **FastAPIUsers**, che implementa le best practice di sicurezza per la gestione delle credenziali.
- **Trasporto Sicuro del Token**: Quando l'utente interagisce tramite l'interfaccia web, per una maggiore sicurezza il token JWT viene memorizzato in un **cookie di tipo HttpOnly**. Questo tipo di cookie non è accessibile tramite JavaScript, mitigando in modo significativo il rischio di furto del token attraverso attacchi di tipo Cross-Site Scripting (XSS).
- **Policy di Accesso**: Sebbene l'architettura sia predisposta per un sistema di autorizzazione basato su ruoli (es. admin, utente, guest), l'implementazione attuale utilizza una policy di accesso binaria: gli utenti non autenticati possono solo leggere i dati, mentre gli utenti autenticati (i partner) hanno i permessi di scrittura per creare progetti e caricare file.

3.13 Riservatezza e Mantenimento dei Dati

Per proteggere la privacy degli utenti, sono state adottate misure specifiche come l'**oscuramento degli indirizzi e-mail** nelle risposte delle API pubbliche. Inoltre, gli utenti hanno il diritto di richiedere la cancellazione del proprio account, che comporta la rimozione sicura dei loro dati personali dal database. Per quanto riguarda il versionamento, i timestamp createdAt e updatedAt nei metadati e la natura immutabile dei file archiviati su Ceph (legati al loro hash) forniscono una tracciabilità storica delle modifiche e garantiscono la conservazione delle versioni originali dei dati conferiti.

4 Descrizione del contenuto del DB ENEA

Il valore di un database scientifico non risiede semplicemente nella quantità di dati che contiene, ma nella sua organizzazione logica, nella ricchezza del suo schema e nella chiarezza con cui le informazioni sono strutturate. Il database ENEA per la piattaforma IEMAP è stato concepito non come un mero archivio, ma come un ecosistema di dati strutturati, progettato per accogliere l'eterogeneità tipica della ricerca sui materiali per l'energia e per abilitare interrogazioni complesse e analisi avanzate. Questo capitolo offre una descrizione approfondita del contenuto del database, analizzandone la struttura, l'organizzazione in collezioni e le modalità di accesso, evidenziando in modo trasparente sia i punti di forza della soluzione implementata sia i suoi limiti intrinseci.

4.1 Tipologie di Dati Memorizzati

Il database ENEA è progettato per ospitare un'ampia gamma di informazioni, che possono essere classificate in quattro categorie principali, ciascuna fondamentale per ricostruire il contesto completo di un'attività di ricerca.

- **Metadati Descrittivi**: Rappresentano il cuore informativo della piattaforma e costituiscono il "passaporto digitale" di ogni esperimento o simulazione. Questi dati contestualizzano e descrivono in dettaglio ogni attività, rendendo le informazioni rintracciabili e interpretabili.

Includono: dettagli sul progetto di appartenenza, descrizioni del processo (metodologia, software, strumentazione), specifiche del materiale (formula chimica, elementi), parametri di input (condizioni operative, impostazioni di calcolo) e proprietà misurate o calcolate (risultati, osservazioni). Questi metadati, essendo strutturati e indicizzati, sono il target primario per le funzionalità di ricerca e filtro della piattaforma.

- **Dati Sperimentali (Time Series e Aggregati)**: La piattaforma è equipaggiata per gestire dati provenienti direttamente da processi sperimentali di laboratorio. L'esempio più significativo, implementato nel corso del progetto, è la raccolta dei dati dal processo di deposizione di film di perovskite. Il database archivia non solo i risultati finali, ma anche i dati grezzi o semi-processati, come le letture sequenziali e temporali dei sensori (temperature, pressioni, velocità di evaporazione). Questo dimostra la capacità del database di fungere da *data logger* strutturato per esperimenti complessi, abilitando analisi a posteriori sui profili temporali dei processi.

- **Riferimenti a Dati Computazionali**: Sebbene i file di output grezzi delle simulazioni computazionali, spesso di grandi dimensioni, siano conservati nel sistema di storage Ceph, il database MongoDB archivia in modo strutturato i metadati essenziali di tali calcoli. Ciò include i parametri di input chiave, le configurazioni utilizzate e, soprattutto, i risultati numerici più importanti estratti dai file di output. Questo approccio ibrido rende i risultati delle simulazioni immediatamente interrogabili, filtrabili e confrontabili, senza la necessità di scaricare e parsare manualmente file di testo complessi e di grandi dimensioni.

- **Dati di Provenienza e Gestione**: Per garantire tracciabilità e sicurezza, il database memorizza anche informazioni di servizio. Ogni record di progetto contiene dati di provenienza (provenance), che includono l'e-mail (oscurata nelle risposte pubbliche) e l'affiliazione del contributore, insieme a timestamp di creazione (createdAt) e ultima modifica (updatedAt). In una collezione separata e isolata (UserAuth), vengono gestite le credenziali degli utenti, garantendo una netta separazione tra i dati scientifici e le informazioni di autenticazione.

4.2 Organizzazione delle Collezioni e Struttura dei Dati

L'architettura del database si basa su MongoDB Enterprise e sfrutta una suddivisione logica in database e collezioni distinti, una best practice per garantire l'ordine, la manutenibilità e le performance del sistema.

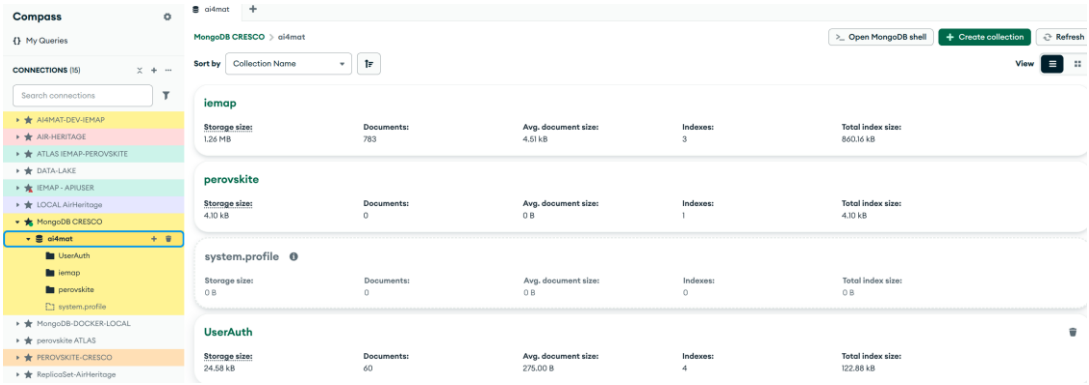

Figura 1. Organizzazione dati in MongoDB, visualizzazione tramite MongoDB Compass

Il database principale per l'intera piattaforma IEMAP è denominato "ai4mat" e contiene due collezioni:

1. iemap
2. UserAuth

iemap, questa è la collezione principale e centrale dell'intera piattaforma, dove risiedono i metadati di tutti i progetti conferiti dai partner. La sua struttura, sebbene flessibile grazie alla natura NoSQL di MongoDB, aderisce a uno schema ben definito, applicato a livello applicativo tramite Pydantic.

UserAuth, è la collezione deputata alla gestione degli account per i partner di progetto e gestita tramite il framework fastapi-users all'interno dell'applicazione FASTAPI.

Un secondo database **Perovskite-CRESCO**, raccoglie i dati sperimentali dei processi di deposizione per celle di perovskite. Si tratta di una funzionalità sviluppata ad hoc con la collaborazione dei colleghi del C.R. Portici che consente lo storage centralizzato dei dati di processo a cui è collegata una applicazione specifica per le funzionalità di data analytics.

Un tipico documento della collezione iemap è un oggetto JSON complesso e nidificato, la cui struttura è di seguito analizzata campo per campo:

- _id (ObjectId): L'identificatore univoco del documento, generato automaticamente da MongoDB.
- iemap_id (String): Un identificatore univoco leggibile e specifico della piattaforma, assegnato al momento della creazione del progetto.
- provenance (Object): Un oggetto che traccia l'origine del dato. Contiene email, affiliation e i timestamp createdAt e updatedAt.

- project (Object): Contiene le informazioni generali sul progetto, come name, label e una description testuale.

- process (Object): Descrive il processo scientifico. Include isExperiment (Boolean), method (es. "DFT"), e un oggetto agent con name (lo strumento/software) e version.

- material (Object): Specifica il materiale studiato, con campi come formula e un array elements estratto automaticamente dalla formula.

- parameters (Array of Objects): Una lista di parametri di input del processo. Ogni oggetto nell'array ha una struttura { "name": String, "value": Number/String, "unit": String }.

- properties (Array of Objects): Una lista delle proprietà/risultati misurati o calcolati, con la stessa struttura dei parametri.

- files (Array of Objects): Una lista che contiene i metadati di tutti i file associati al progetto. Ogni oggetto include hash, name, extension, size e timestamp. Questo array è il ponte che collega il documento di metadati in MongoDB ai file fisici su Ceph.

Valutazione Critica della Soluzione:

- **Punti di Forza (Pro):**

    - **Flessibilità**: La struttura a documenti JSON nidificati è eccezionalmente adatta a rappresentare dati scientifici complessi e gerarchici. Permette di aggiungere nuovi campi o strutture senza richiedere costose e rigide migrazioni di schema, una caratteristica essenziale in un contesto di ricerca in evoluzione.

    - **Coerenza con lo Stack Tecnologico**: L'uso di JSON nativo si integra perfettamente con le moderne tecnologie web (JavaScript/VueJS nel front-end) e con Python (che mappa nativamente i dizionari a JSON), riducendo la complessità e l'overhead di serializzazione/deserializzazione.

- **Limiti e Margini di Miglioramento (Contro):**

    - **Rischio di Incoerenza**: La flessibilità di NoSQL, se non governata, può portare a dati inconsistenti. Nella soluzione IEMAP, questo rischio è fortemente mitigato dalla validazione rigorosa tramite Pydantic a livello applicativo. Tuttavia, qualsiasi interazione con il database che bypassi l'API (es. per scopi amministrativi) richiede la massima disciplina per non introdurre dati non conformi.

    - **Complessità delle Query**: Interrogare campi profondamente nidificati o eseguire aggregazioni complesse su questa struttura può essere sintatticamente più complesso e potenzialmente meno performante rispetto a un modello relazionale normalizzato.

- **Mancanza di Semantica Nativa**: Come evidenziato nella sezione sulla FAIRness, MongoDB non supporta nativamente modelli semantici come RDF o linguaggi di interrogazione come SPARQL. La "semantica" è implicita nella struttura del documento ma non è formalmente definita a livello di database, limitando l'interoperabilità con strumenti del web semantico.

- **4.2.2 Database perovskite - Collezioni Specializzate**

Per gestire il caso d'uso specifico della deposizione di perovskite, è stato creato un database separato, dimostrando la capacità della piattaforma di isolare logicamente dati di progetti diversi. Questo database contiene due collezioni principali:

- **Collezione sensors_readings (Time Series)**: Questa non è una collezione standard, ma una **Time Series (TS) Collection**, una feature specializzata di MongoDB ottimizzata per l'ingestione, l'archiviazione e l'interrogazione ad alte prestazioni di dati con timestamp. Questa scelta architetturale è un notevole punto di forza, poiché riduce l'impronta su disco, migliora l'efficienza delle query temporali e semplifica la gestione di dati di sensori ad alta frequenza.

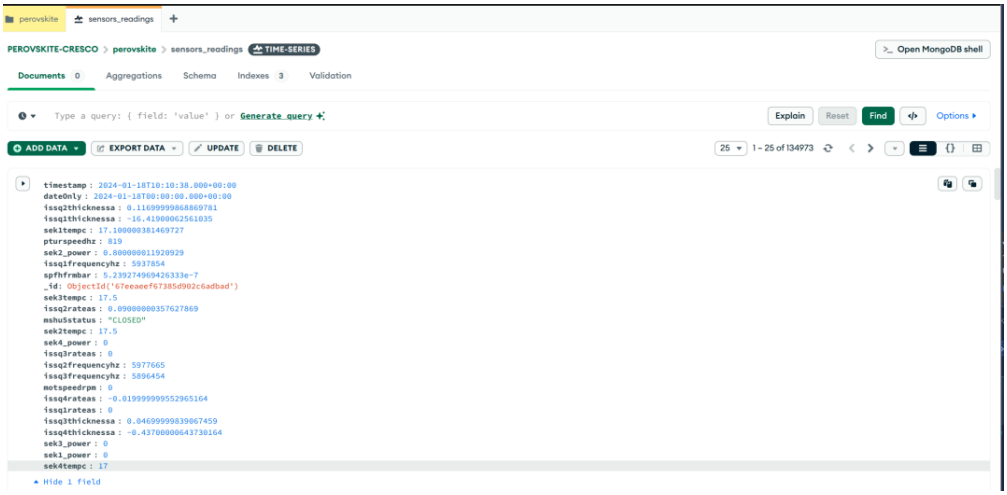

Figura 2. Collezione TS per dati Perovskite, visualizzazione MongoDB Commpass

- **Collezione source_events**: Questa collezione agisce da complemento alla precedente. Mentre sensors_readings contiene i dati grezzi, source_events archivia dati sintetici, statistici o eventi aggregati derivati dai processi di deposizione. Questo permette di eseguire query rapide per ottenere sommari e viste d'insieme senza dover processare ogni volta l'intera mole di dati time-series.

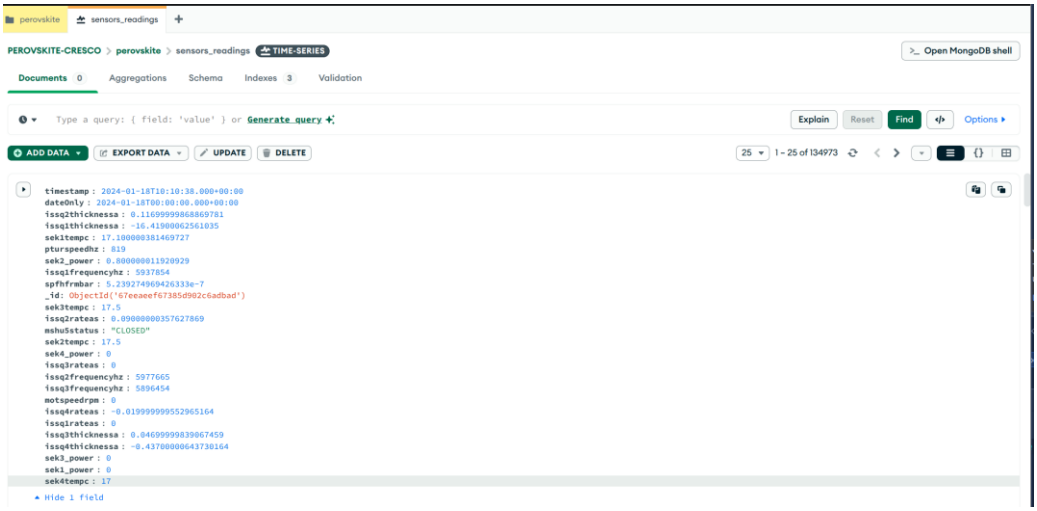

Figura 3. Collezione di supporto per dati da processo Perovskite, visualizzazione MongoDB Compass

Valutazione Critica della Soluzione:

- **Punti di Forza (Pro):**
    - **Ottimizzazione delle Performance**: L'uso di una TS Collection è la scelta tecnologicamente più appropriata e performante per i dati sperimentali sequenziali, dimostrando una progettazione attenta e consapevole delle feature avanzate del database.
    - **Separazione delle Competenze**: La strategia di separare i dati grezzi (time series) dai dati aggregati (eventi) è una best practice consolidata nell'ingegneria dei dati, che garantisce performance ottimali per entrambe le tipologie di interrogazione (analisi di dettaglio vs. viste di sintesi).
- **Limiti e Margini di Miglioramento (Contro):**
    - **Scarsa Generalizzabilità dello Schema**: Lo schema di queste collezioni è altamente specifico e personalizzato per il processo di deposizione di perovskite. Sebbene la soluzione sia eccellente per questo caso d'uso, non è direttamente generalizzabile ad altri tipi di esperimenti. L'onboarding di un nuovo processo sperimentale sulla piattaforma richiederebbe un'analisi dedicata e la progettazione di un nuovo schema su misura, evidenziando una sfida nella creazione di una soluzione "universale" per i dati di laboratorio.
- **4.3 Accessibilità del Database: Modalità per Utenti Interni ed Esterni**

L'accesso al database ENEA è governato da un principio di sicurezza fondamentale: **nessun accesso diretto dall'esterno**. Tutte le interazioni sono mediate da strati di astrazione che garantiscono sicurezza, controllo e manutenibilità.

- **Accesso Esterno (Partner e Pubblico)**: L'unico modo per utenti esterni di interagire con i dati è attraverso la **REST API**. Questo è un punto architetturale cruciale. L'API funge da "gatekeeper":

    1. **Protegge** il database da esposizioni dirette sulla rete, riducendo drasticamente la superficie di attacco.

    2. **Applica la logica di business**, come le regole di validazione dei dati e i controlli di autorizzazione (permessi di scrittura solo per i partner).

    3. **Disaccoppia i client dall'implementazione fisica**: il front-end o il modulo Python non "conoscono" la tecnologia del database sottostante. Ciò significa che in futuro il database potrebbe essere sostituito (es. con un'altra tecnologia) senza dover modificare le applicazioni client, purché l'interfaccia API rimanga la stessa.

- **Accesso Interno (Amministratori e Servizi)**: L'accesso diretto al cluster MongoDB è consentito solo a un numero ristretto di amministratori di sistema e ai servizi applicativi del back-end in esecuzione sulla rete interna di ENEA. Per le operazioni di gestione, monitoraggio e manutenzione, gli amministratori si avvalgono di strumenti enterprise:

    - **MongoDB Compass**: Un'interfaccia grafica (GUI), questa OS, che permette di visualizzare i dati, eseguire query manuali, analizzare le performance e gestire gli indici in modo visuale.

    - **MongoDB Ops Manager**: Una console di gestione centralizzata utilizzata per le operazioni più critiche, come la configurazione del cluster, il monitoraggio delle performance in tempo reale, la gestione degli allarmi e, soprattutto, l'orchestrazione delle procedure di backup e ripristino.

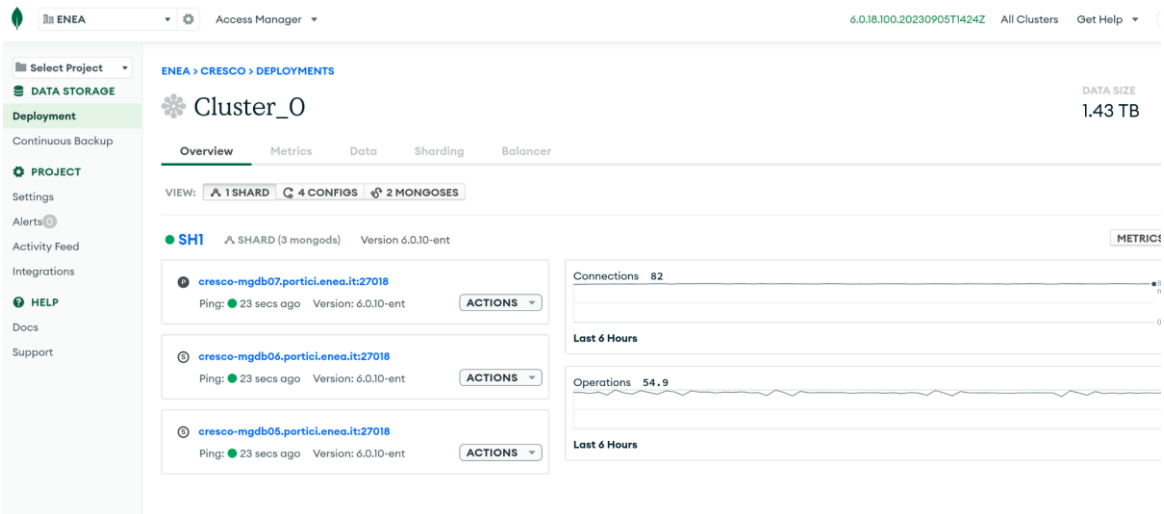

Figura 4. Dashboard di Ops Manager per il monitoraggio del cluster MongoDb in CRESCO

Valutazione Critica della Soluzione:

- **Punti di Forza (Pro):**
    - **Sicurezza**: L'approccio API-first è una best practice consolidata per la sicurezza delle applicazioni web.
    - **Manutenibilità e Scalabilità**: L'architettura a servizi disaccoppiati è intrinsecamente più manutenibile e scalabile rispetto a un'applicazione monolitica con accesso diretto al database.
    - **Gestione Enterprise**: L'uso di strumenti come Ops Manager eleva la gestione del database da un livello artigianale a uno di tipo enterprise, garantendo affidabilità e controllo.
- **Limiti e Margini di Miglioramento (Contro):**
    - **Limitazioni per l'Analisi Esplorativa**: Il principale compromesso di questo approccio è che gli utenti finali, anche quelli più esperti, non possono eseguire query arbitrarie e complesse direttamente sul database. Sono limitati alle funzionalità di interrogazione esposte dall'API. Questo può rappresentare un ostacolo per l'analisi esplorativa dei dati, che spesso richiede query non prevedibili a priori. Si tratta, tuttavia, di un trade-off deliberato, dove la sicurezza, la stabilità e la governance del dato sono state prioritizzate rispetto alla completa libertà di interrogazione. Si potrebbe ovviare a tale limitazione sviluppando una soluzione AI che tramite un server MCP (Model Context Protocol) consente di sfruttare un modello LLM OS con un opportuno prompt per eseguire anche query complesse, ma sempre di sola lettura per garantire la protezione dei dati, senza dover conoscere lo schema dei dati e/o l'Aggregation Framework di MongoDB.

5 FAIRness del DB ENEA

Nell'ambito della ricerca scientifica moderna, la creazione di un database non può prescindere da una valutazione rigorosa della sua capacità di rendere i dati effettivamente utili e valorizzabili per la comunità. A tal fine, i principi FAIR si sono affermati come il paradigma di riferimento internazionale per la gestione dei dati di ricerca. Essi non costituiscono uno standard tecnologico rigido, ma un insieme di linee guida qualitative che mirano a massimizzare il valore dei dati, garantendo che siano Trovabili, Accessibili, Interoperabili e Riusabili (**Findable, Accessible, Interoperable, Reusable**), sia per gli esseri umani che per i sistemi computazionali. Questo capitolo fornisce un'introduzione dettagliata a ciascun principio, descrive la metodologia adottata per valutare la conformità del database ENEA e presenta un'analisi critica dei risultati ottenuti, evidenziando i punti di forza della soluzione implementata e delineando una chiara roadmap per i futuri miglioramenti.

5.1 Principi FAIR: Un Paradigma per i Dati della Ricerca

I principi FAIR sono stati formulati per guidare la gestione e la curatela dei dati scientifici, spostando l'enfasi dalla semplice archiviazione alla creazione di un vero e proprio patrimonio informativo, pronto

per essere scoperto, integrato e riutilizzato in contesti di ricerca nuovi e imprevisti. Comprendere il significato profondo di ciascun principio è fondamentale per apprezzare le scelte architetturali del database IEMAP.

F - Findable (Trovabili): Il primo passo per poter riutilizzare un dato è trovarlo. Il principio "Findable" va oltre la semplice ricerca testuale. Richiede che i dati siano descritti da metadati ricchi e che sia ai dati che ai metadati vengano assegnati identificatori univoci a livello globale e persistenti (es. un DOI - Digital Object Identifier). Questo garantisce che una risorsa possa essere citata e ritrovata in modo non ambiguo nel tempo, anche se la sua posizione fisica dovesse cambiare. Inoltre, i metadati devono essere indicizzati in una risorsa ricercabile, come un catalogo o un portale pubblico, quale è la piattaforma IEMAP stessa.

A - Accessible (Accessibili): Una volta trovati, i dati devono essere recuperabili attraverso un protocollo di comunicazione standard, aperto, gratuito e universalmente implementabile. Questo principio non implica che i dati debbano essere necessariamente "aperti a tutti", ma che le modalità tecniche per ottenerli siano chiare e basate su standard condivisi. Nel contesto della piattaforma IEMAP, questo principio è pienamente realizzato attraverso un'architettura API-first. L'accesso ai dati non avviene tramite un download diretto di file da un repository, ma attraverso l'interazione programmatica con un'interfaccia REST API. Questo approccio garantisce che:

- **Il protocollo di recupero è standardizzato**: Le REST API sono lo standard de facto per l'interscambio di dati sul web. Qualsiasi applicazione o script moderno può interagire con esse utilizzando metodi HTTP convenzionali (GET, POST, ecc.).

- **Il formato dei dati è interoperabile**: I dati interrogati tramite l'API vengono restituiti in formato **JSON (JavaScript Object Notation)**, un formato testuale leggero, leggibile dall'uomo e facilmente processabile da qualsiasi linguaggio di programmazione, garantendo un'accessibilità a livello macchina.

Il principio "Accessible" copre in modo esplicito anche i casi in cui l'accesso è differenziato e controllato. Nella piattaforma IEMAP:

- L'accesso in lettura (GET) è pubblico: Chiunque può interrogare l'API per consultare i metadati dei progetti senza necessità di autenticazione.

- L'accesso in scrittura e modifica (operazioni CUD - Create, Update, Delete) è protetto e richiede un'autenticazione specifica. La procedura per ottenere i permessi è ben documentata: un partner autorizzato deve autenticarsi tramite l'API per ricevere un JWT (JSON Web Token), da includere nelle richieste successive per eseguire operazioni di modifica.

Infine, il protocollo HTTPS agisce come un livello di trasporto sicuro che incapsula tutte le comunicazioni con l'API, garantendo la confidenzialità e l'integrità dei dati durante il loro trasferimento.

In sintesi, l'accessibilità in IEMAP non è solo garantita dalla sicurezza del trasporto (HTTPS), ma è definita in modo più profondo dalla disponibilità di un'interfaccia API standard (REST) che fornisce dati

in un formato universale (JSON) e gestisce in modo trasparente e documentato sia l'accesso pubblico che quello autenticato (JWT). Le "regole del gioco" per accedere ai dati sono chiare, standard e comprensibili sia per un utente che per una macchina.

I - Interoperable (Interoperabili): Questo è forse il principio più tecnico e ambizioso. L'interoperabilità è la capacità dei dati di essere combinati e processati in modo automatico da sistemi computazionali, anche se provenienti da fonti diverse. Per raggiungere questo obiettivo, i dati devono "parlare una lingua comune". Ciò si ottiene attraverso:

- L'uso di formati di dati standard e aperti (es. JSON, CSV, come in IEMAP).
- L'adozione di vocabolari, tassonomie e ontologie condivise per descrivere i metadati. Un'ontologia (come quella in formato OWL sviluppata per IEMAP) definisce formalmente i concetti e le relazioni tra di essi (es. definisce cos'è un "Progetto" e come si relaziona a un "Materiale"), eliminando le ambiguità semantiche e permettendo a un computer di "comprendere" la struttura del dato.

R - Reusable (Riusabili): La riusabilità è l'obiettivo ultimo dei principi FAIR. Affinché un dato possa essere effettivamente riutilizzato in studi futuri, deve essere descritto in modo così ricco e preciso da permettere a un altro ricercatore, o a un sistema computazionale, di comprenderne appieno il contesto, l'origine e l'affidabilità. Nella piattaforma IEMAP, questo principio non è un concetto astratto, ma è supportato da specifiche implementazioni a livello di schema del database e di policy di gestione.
In particolare, la riusabilità è garantita da tre pilastri fondamentali implementati nel sistema:

1. **Metadati Ricchi e Contestualizzati**: La condizione primaria per il riutilizzo è la comprensione. Lo schema del database MongoDB non si limita a memorizzare un risultato, ma impone la raccolta di un ricco contesto descrittivo per ogni dato.
   Ogni progetto archiviato deve obbligatoriamente includere:
   - Dettagli sul processo utilizzato (process), specificando il metodo (es. "DFT", "Cyclic voltammetry") e l'agente (lo strumento o il software con la sua versione).
   - La definizione precisa del materiale (material), inclusa la sua formula chimica.
   - Un elenco strutturato di parametri di input (parameters) e proprietà misurate (properties), ciascuno descritto da una tripla {"name", "value", "unit"}.

   Questa struttura garantisce che un valore numerico sia sempre accompagnato dalla sua unità di misura, un dettaglio fondamentale per evitare errori di interpretazione e consentire il riutilizzo in nuove analisi.

2. **Provenienza Dettagliata e Tracciabile**: Per valutare l'affidabilità di un dato e poterlo citare correttamente, è indispensabile conoscerne l'origine. Il database IEMAP implementa questo requisito attraverso l'oggetto **provenance** presente in ogni singolo documento di progetto. Questo oggetto registra in modo automatico e non modificabile:
   - L'affiliazione e l'e-mail del contributore che ha caricato il dato.
   - La data e ora di creazione (createdAt) e di ultima modifica (updatedAt) del record di metadati.
   - Questa tracciabilità assicura che ogni dato sia sempre associato a una fonte verificabile e a un contesto temporale preciso, elementi essenziali per il suo riutilizzo in contesti scientifici rigorosi.

3. **Aderenza a Standard e Vocabolari Formali**:

   La riusabilità dei dati è massimizzata quando essi sono descritti utilizzando linguaggi e vocabolari condivisi e comprensibili dalle macchine. La piattaforma IEMAP ha compiuto un passo decisivo in questa direzione attraverso la definizione e pubblicazione di un'ontologia formale. **È stata sviluppata un'ontologia specifica per il progetto in formato OWL (Web Ontology Language)**, derivandone la struttura direttamente dallo schema dei dati di MongoDB. Questo modello formale definisce in modo non ambiguo le entità chiave del dominio (come Project, Process, Material, Parameter) e le relazioni che intercorrono tra di esse. Per garantire la persistenza, la citabilità e l'accesso pubblico, **l'ontologia è stata pubblicata sul repository aperto Zenodo**, un passo cruciale per la conformità ai principi FAIR. A livello operativo, questa standardizzazione terminologica viene applicata fin dal momento dell'inserimento dei dati. I vocabolari controllati presenti nei menù a discesa del template Excel sono l'applicazione pratica di questo modello, garantendo che tutti i partner utilizzino gli stessi termini per descrivere concetti equivalenti. Questa combinazione di un modello semantico formale (l'ontologia OWL) e la sua applicazione pratica (i vocabolari controllati) è cruciale: assicura che esperimenti simili condotti da partner diversi siano descritti in modo coerente, rendendo i dati aggregabili, confrontabili e quindi significativamente più riutilizzabili per analisi su larga scala. **La roadmap futura prevede di arricchire ulteriormente questo lavoro, versionando l'ontologia su GitHub e mappandola a standard internazionali come EMMO e ChEB**I.

5.2 Metodologia di Valutazione della FAIRness

Per tradurre i principi FAIR da concetti astratti a metriche concrete, è stata adottata una metodologia di valutazione semi-quantitativa. Questo approccio si basa sulla scomposizione di ciascuno dei quattro principi guida (Findable, Accessible, Interoperable, Reusable) in un insieme di criteri specifici e misurabili, applicati non alla piattaforma come entità monolitica, ma singolarmente alle sue tre componenti architetturali chiave:

1. Il Database (MongoDB): il cuore della conservazione dei dati.
2. L'Interfaccia Programmatica (REST API): il canale primario per l'accesso macchina-macchina.
3. L'Interfaccia Utente (Sito Web): il canale principale per l'interazione umana.

Ad ogni criterio è stato assegnato un punteggio per ciascuna componente, riflettendo il grado di aderenza raggiunto. Questo processo ha permesso di ottenere una visione granulare e onesta della maturità FAIR del sistema, evidenziando come diverse parti dell'architettura contribuiscano in modo differente al raggiungimento degli obiettivi. I risultati di questa analisi dettagliata sono sintetizzati nella tabella seguente, che riporta i punteggi aggregati per ciascun principio e il punteggio di FAIRness complessivo per ogni componente.

Componente	Findable (6)	Accessible (6)	Interoperable (6)	Reusable (6)	Totale (24)	FAIRness (%)
MongoDB	5/6	6/6	3/6	5/6	19/24	79.2%
REST API	5/6	6/6	4/6	6/6	21/24	87.5%
Sito Web	4/6	5/6	3/6	4/6	16/24	66.7%

Tabella 1. FAIRness Score - punteggio sintetico raggiungimento principi FAIR

Dettaglio della Metodologia di Punteggio

Per garantire una valutazione oggettiva e riproducibile, è stata definita una rubrica di valutazione basata su una scala di punteggio a 3 livelli per ogni criterio specifico:

- **0 punti: Criterio non soddisfatto o non implementato.**
- **1 punto: Criterio parzialmente soddisfatto o implementato in modo indiretto.**
- **2 punti: Criterio pienamente soddisfatto, con un'implementazione diretta e robusta.**

Spiegazione del Denominatore (il "/6")

Ogni principio FAIR (Findable, Accessible, Interoperable, Reusable) è stato scomposto in tre criteri di valutazione principali. Di conseguenza, il punteggio massimo teorico che una componente può ottenere per un singolo principio è dato da:

3 criteri × 2 punti (punteggio massimo per criterio) = 6 punti

Questo calcolo spiega perché tutti i punteggi nella tabella di sintesi sono espressi come una frazione di 6. Il denominatore rappresenta il punteggio ideale di piena conformità.

Analisi di un Caso Specifico: Perché MongoDB ha 5/6 per "Findable"?

Analizziamo nel dettaglio come è stato calcolato il punteggio 5/6 per la componente MongoDB relativamente al principio Findable, basandoci sui criteri specifici elencati nella Tabella 3 del documento di analisi.

I tre criteri per "Findable" sono:

Identificatore persistente (iemap_id): Valuta se ai dati è assegnato un ID univoco e durevole.
Metadati ricercabili via Python / Web: Valuta se i metadati sono archiviati in modo da permettere interrogazioni complesse.
Documentazione di discovery (Swagger): Valuta la presenza di documentazione che faciliti la scoperta automatica delle risorse.
L'attribuzione del punteggio per MongoDB su questi tre criteri è la seguente:

Criterio 1: Identificatore persistente → Punteggio: 2/2
Motivazione: Il database implementa nativamente un campo iemap_id per ogni progetto. Questo identificatore è gestito a livello applicativo per garantire unicità e persistenza, soddisfacendo pienamente il requisito.

Criterio 2: Metadati ricercabili → Punteggio: 2/2
Motivazione: MongoDB è un database NoSQL orientato ai documenti, la cui funzione primaria è proprio quella di archiviare e indicizzare dati semi-strutturati (i metadati) per consentire interrogazioni complesse ed efficienti. La capacità di ricerca è una caratteristica intrinseca e pienamente sfruttata, quindi il criterio è completamente soddisfatto.

Criterio 3: Documentazione di discovery → Punteggio: 1/2
Motivazione: La documentazione di discovery tramite Swagger UI è una caratteristica intrinseca della REST API, non del database MongoDB in sé. Il database, come componente isolato, non possiede un meccanismo nativo di auto-documentazione per la scoperta da parte di client esterni. Tuttavia, poiché i suoi dati sono resi "discoverabili" attraverso l'API documentata da Swagger, si considera che il criterio sia parzialmente o indirettamente soddisfatto dal punto di vista del database. Non riceve il punteggio pieno perché non è una sua funzionalità nativa, ma ne beneficia indirettamente.

Calcolo del Punteggio Finale:
Sommando i punteggi ottenuti per ciascun criterio, si ottiene il numeratore finale:

2(ID)+2(Ricerca)+1(Doc)=5 punti

Ecco spiegato perché il punteggio finale per il database MongoDB nella categoria Findable è 5/6. Questa stessa logica granulare, basata sulla valutazione di criteri specifici e sull'analisi del ruolo diretto o indiretto di ogni componente, è stata applicata per calcolare ogni singolo valore presente nella tabella di sintesi, garantendo una valutazione coerente e giustificabile in ogni sua parte.

Analisi e Interpretazione dei Punteggi

L'analisi dei punteggi rivela un quadro coerente con le scelte progettuali della piattaforma e offre spunti di riflessione importanti.
REST API (Punteggio più alto: 87.5%): L'interfaccia API emerge come la componente più conforme ai principi FAIR, ottenendo il punteggio più elevato. Questo risultato non è casuale: l'API è per sua natura un'interfaccia macchina-macchina, progettata esplicitamente per la condivisione strutturata di dati. Eccelle in Accessibilità (6/6), grazie all'uso di protocolli standard e autenticazione chiara, e in Riusabilità (6/6), poiché espone i dati in un formato (JSON) e con metadati di provenienza che ne facilitano il riutilizzo programmatico. Il suo punteggio in Interoperabilità (4/6) è superiore a quello del database stesso, poiché l'API può agire da mediatore, esponendo i dati in un formato più standardizzato (JSON Schema) di quanto non sia nativamente nel DB.
Database MongoDB (Punteggio intermedio: 79.2%): Il database, pur essendo il repository primario, ottiene un punteggio leggermente inferiore. Mantiene un'eccellente Accessibilità (6/6), essendo il sistema a cui l'API si connette in modo standard, e una buona Riusabilità (5/6) grazie allo schema ricco di metadati di provenienza. Il punto debole, che ne abbassa significativamente il punteggio, è l' Interoperabilità (3/6). Come discusso in precedenza, questo è un riflesso diretto della scelta tecnologica: MongoDB non è un database semantico nativo. Sebbene sia stata definita un'ontologia OWL esterna, il database stesso non la utilizza per l'interrogazione o la validazione, limitando la sua interoperabilità a livello macchina. Sito Web (Punteggio più basso: 66.7%): L'interfaccia utente ottiene, come prevedibile, il punteggio più basso. Questo perché il sito web è ottimizzato per l'interazione umana, non per quella programmatica. Sebbene renda i dati efficacemente Trovabili (4/6) e Accessibili

(5/6) per un utente umano, la sua capacità di servire dati in modo Interoperabile (3/6) e Riusabile (4/6) per un sistema automatico è intrinsecamente limitata. Estrarre dati da una pagina HTML (scraping) è un processo fragile e non standard, a differenza dell'interrogazione di un'API.

In conclusione, questa valutazione numerica conferma la validità dell'approccio API-first adottato dal progetto IEMAP. L'API non è solo un canale di accesso, ma è la componente che più di ogni altra realizza la promessa della FAIRness, fungendo da ponte standardizzato e di alta qualità verso il patrimonio di dati conservato nel database

5.3 Risultati della Valutazione e Analisi Critica

L'applicazione della metodologia di valutazione FAIR ha prodotto un quadro dettagliato e trasparente del livello di conformità del database ENEA. I risultati, qui presentati e analizzati, non indicano un semplice punteggio finale, ma delineano un profilo strategico con aree di eccellenza consolidata e altre di potenziale evolutivo.

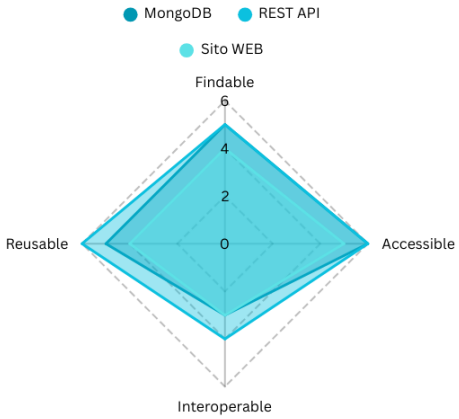

Figura 5. Diagramma sintetico soddisfacimento FAIRness per piattaforma IEMAP

L'analisi critica dei punteggi, suddivisa per principio, offre la seguente visione d'insieme.
Findable e Accessible (Punti di Forza Consolidati): Il database ottiene punteggi molto elevati in queste due aree fondamentali, che rappresentano il primo gradino per ogni interazione con i dati.
La trovabilità è garantita dall'assegnazione di un identificatore persistente (iemap_id) a ogni progetto, e dalla disponibilità di metadati ricercabili sia tramite il front-end web che via API.
L' accessibilità è pienamente realizzata attraverso l'uso del protocollo standard HTTPS per tutte le comunicazioni e un sistema di autenticazione basato su JWT ben definito, che regola l'accesso alle risorse in modo sicuro. La presenza di una documentazione interattiva (Swagger UI) per la REST API chiarisce ulteriormente le modalità di accesso, soddisfacendo pienamente i requisiti del principio A.
Interoperable (Un Passo Decisivo Compiuto): L'analisi dell'interoperabilità è particolarmente significativa e merita un'attenzione speciale. Sebbene il punteggio sintetico sia inferiore rispetto agli altri principi, esso non riflette una mancanza, ma lo stato di avanzamento di una strategia semantica già in atto. Il risultato più importante in quest'area è che non ci si è limitati a formati di base, ma è stata progettata e implementata un'ontologia formale in formato OWL. Questa ontologia, derivata direttamente dallo schema dei dati di MongoDB, definisce in un linguaggio standard e comprensibile dalle macchine i concetti e le relazioni del dominio. Ancor più importante, l'ontologia è stata

pubblicata sul repository aperto Zenodo, rendendo il modello dei dati stesso un artefatto citable e FAIR. **La valutazione critica evidenzia quindi un punto cruciale: il lavoro di formalizzazione semantica è stato completato. Il margine di miglioramento, che giustifica il punteggio non massimale, risiede nel fatto che questa interoperabilità semantica non è ancora completamente operativa all'interno della piattaforma. L'ontologia, al momento, agisce come un descrittore esterno e non è ancora integrata nei meccanismi di validazione automatica o di interrogazione del database**. La base dell'interoperabilità è comunque garantita dall'uso di formati standard come JSON per lo scambio di dati.
Reusable (Buon Livello di Conformità): Anche la Riusabilità, obiettivo ultimo della FAIRness, raggiunge un buon livello di conformità. Questo risultato è il frutto diretto di diverse scelte implementative:

- Metadati Ricchi e Contestualizzati: Lo schema del database impone la raccolta di un contesto descrittivo dettagliato, essenziale per permettere a terzi di comprendere e riutilizzare i dati.
- Provenienza Tracciata: Ogni documento include un oggetto provenance che traccia l'autore e i timestamp di creazione e modifica, permettendo di valutare l'origine e l'affidabilità del dato.
- Versioning e Licenze: Il sistema supporta il versionamento automatico dei documenti e, aspetto fondamentale per la riusabilità dell'intero ecosistema, i componenti software come il modulo
- iemap-mi sono rilasciati con una licenza d'uso chiara e aperta (MIT). Un margine di miglioramento futuro consisterà nell'associare una licenza esplicita a ogni singolo dataset, oltre che al software.

In conclusione, la valutazione FAIRness del database ENEA mostra un sistema maturo, con eccellenti performance su reperibilità e accessibilità e una solida base per la riusabilità. Il lavoro pionieristico svolto sulla definizione e pubblicazione dell'ontologia posiziona la piattaforma in una traiettoria strategica per raggiungere nel prossimo futuro il più alto livello di interoperabilità semantica.

5.4 Analisi Comparativa: la Metodologia IEMAP e il Panorama degli Strumenti FAIR

L'approccio di autovalutazione con indici sintetici adottato per la piattaforma IEMAP si inserisce in un panorama consolidato di strumenti e framework per la misurazione della FAIRness. Per comprendere appieno i punti di forza e i limiti della metodologia scelta, è utile analizzare le principali soluzioni software esterne e confrontarle con il processo di audit interno condotto sul progetto.

Analisi delle Principali Soluzioni Software per la Valutazione FAIRness
Gli strumenti per la valutazione FAIR si possono suddividere in diverse categorie, ciascuna con un approccio e un'applicabilità specifici.

Piattaforme di Valutazione Automatica
Questi strumenti sono progettati per fornire un punteggio numerico e oggettivo interrogando programmaticamente una risorsa digitale accessibile via web.

- **F-UJI (FAIRsFAIR, https://w3id.org/fuji)**
  Come funziona: È un servizio web che accetta un Identificatore Persistente (PID, come un DOI) di un dataset. F-UJI visita la pagina associata all'ID, ne analizza i metadati (sia a livello HTML che tramite JSON-LD o altri formati strutturati) e li testa contro un set di 17 metriche standardizzate. Il risultato è un report dettagliato e un punteggio numerico per ogni principio FAIR.
  Pro: Fornisce una valutazione oggettiva, rapida e riproducibile. Essendo basato su metriche

standard, permette di confrontare diversi repository in modo omogeneo.
Contro: Può testare solo risorse web pubbliche e non può analizzare componenti interne come un database. Inoltre, verifica la presenza e la forma tecnica dei metadati, ma non può valutarne la qualità o la rilevanza semantica (un campo "licenza" con scritto "sconosciuto" potrebbe superare il test tecnico).

- FAIR Evaluator (GO FAIR, https://fairaware.dans.knaw.nl/)
  Come funziona: È un framework più generico che esegue test di conformità basati su una collezione di "Indicatori di Maturità" (Maturity Indicators). La sua forza risiede nella flessibilità: diverse comunità possono definire e registrare i propri set di test, rendendolo adattabile a specifici domini di ricerca.
  Pro: Altamente flessibile e estensibile, permette valutazioni personalizzate e specifiche per un dato dominio scientifico.
  Contro: Richiede una maggiore configurazione e comprensione tecnica rispetto a F-UJI. La sua flessibilità può rendere più difficile il confronto tra valutazioni basate su set di metriche differenti.

- FAIRshake (https://fairaware.dans.knaw.nl/)
  Come funziona: Una piattaforma che si concentra sulla valutazione collaborativa. Gli utenti possono definire delle "rubriche" (checklist di metriche) e applicarle a risorse digitali, che vengono poi visualizzate tramite "badge" grafici che ne certificano il livello di aderenza.
  Pro: Approccio visuale e intuitivo basato sui badge. Promuove il coinvolgimento della comunità nella definizione di cosa è "FAIR" in un dato contesto.
  Contro: La valutazione può essere più soggettiva, in quanto dipende dalle rubriche create dagli utenti. Meno standardizzato rispetto a F-UJI.

Strumenti di Autovalutazione Guidata
Questi strumenti sono pensati per assistere l'utente in una valutazione manuale o semi-automatica, spesso tramite questionari interattivi.

- FAIR-Aware (EUDAT)
  Come funziona: È uno strumento online sotto forma di questionario guidato. Pone all'utente 10 domande per aiutarlo a comprendere il livello di "FAIR-readiness" dei propri dati prima della loro pubblicazione, fornendo consigli pratici.
  Pro: Eccellente strumento formativo e di sensibilizzazione. Semplice da usare e non richiede competenze tecniche.
  Contro: Il risultato è basato interamente sull'autodichiarazione dell'utente e non effettua alcuna verifica tecnica. Non fornisce un punteggio quantitativo dettagliato.

Architetture "FAIR by Design"
Queste soluzioni non sono solo validatori, ma framework software per pubblicare dati in un modo che sia intrinsecamente FAIR.

- FAIR Data Point (FDP)
  Come funziona: È una specifica e un'implementazione software per creare un endpoint di metadati per un repository. Un FDP espone i metadati secondo un modello predefinito e tramite un'API, rendendoli immediatamente accessibili e interrogabili dalle macchine.
  Pro: Approccio proattivo ("FAIR by design"). Garantisce che i metadati siano nativamente leggibili dalle macchine, facilitando la creazione di ecosistemi di dati federati.

Contro: Non è uno strumento di valutazione per un sistema esistente, ma richiede l'adozione di una specifica architettura software, che può rappresentare un onere implementativo significativo.

Relazione e Confronto con l'Approccio IEMAP
L'approccio di autovalutazione e calcolo di indici sintetici utilizzato per la piattaforma IEMAP si configura come una soluzione ibrida e pragmatica, che trae ispirazione da diverse delle metodologie sopra descritte. Somiglianza con i Framework Manuali: La metodologia IEMAP è, nella sua essenza, un'applicazione del principio dei framework di autovalutazione come il RDA FAIR Data Maturity Model. Invece di affidarsi a un test automatico esterno, è stata creata una rubrica di valutazione interna e su misura, basata su criteri specifici derivati dai principi FAIR ma adattati al contesto tecnologico della piattaforma (MongoDB, REST API, ecc.). Questo ha permesso di condurre un' analisi profonda e contestualizzata, in grado di valutare componenti interne (come lo schema del database) che sarebbero inaccessibili a qualsiasi strumento automatico esterno.
Somiglianza con gli Strumenti Automatici: A differenza di una semplice checklist qualitativa, il processo di valutazione IEMAP culmina nell'attribuzione di un punteggio numerico e di un indice percentuale sintetico. In questo, emula l'obiettivo degli strumenti automatici come F-UJI, ovvero fornire una metrica quantitativa che permetta di riassumere rapidamente il livello di conformità e di monitorare i progressi nel tempo.
Conclusione del Confronto:
L'approccio IEMAP può essere definito come un audit interno strutturato. Il suo principale vantaggio è la capacità di fornire una valutazione granulare e consapevole del contesto, spiegando il "perché" dietro ogni punteggio. Il suo limite intrinseco, d'altra parte, è una potenziale mancanza di oggettività e standardizzazione rispetto a un benchmark esterno come F-UJI.
In definitiva, la metodologia scelta è stata la più appropriata per la fase di sviluppo e validazione interna del progetto, fornendo un'analisi profonda e attuabile. Un passo futuro logico per la piattaforma IEMAP sarà quello di complementare questo audit interno utilizzando uno strumento automatico come F-UJI per validare le risorse pubbliche esposte dall'API. Questo permetterebbe di confrontare i risultati interni con un benchmark esterno e standardizzato, ottenendo una visione della FAIRness completa sia dall'interno che dall'esterno.

5.5 Sintesi e Roadmap per il Miglioramento della FAIRness

In conclusione, il database ENEA per la piattaforma IEMAP dimostra un livello di FAIRness complessivamente buono, con punte di eccellenza negli ambiti "Findable" e "Accessible". La soluzione rappresenta un eccellente compromesso tra pragmatismo tecnologico (l'uso di uno stack scalabile e diffuso come MongoDB) e l'adesione a principi di gestione dei dati avanzati. I risultati dell'analisi non rappresentano un punto di arrivo, ma la base per una roadmap di miglioramento continuo. Le attività future, come delineato nel piano di progetto, si concentreranno strategicamente sull'aumento dell'interoperabilità, capitalizzando il lavoro già svolto con la creazione dell'ontologia OWL. La roadmap prevede:

1. **Mappatura a Vocabolari Standard**: Arricchire l'ontologia IEMAP creando dei collegamenti formali (mapping) a vocabolari standard internazionali nel dominio dei materiali e della chimica, come EMMO (European Materials Modelling Ontology), ChEBI (Chemical Entities of Biological Interest) e QUDT (Quantities, Units, Dimensions and Types). Questo passo aumenterebbe esponenzialmente l'interoperabilità semantica dei dati IEMAP.

2. Sperimentazione con Database Semantici: Avviare attività di ricerca e sviluppo per esplorare

l'integrazione della piattaforma con database semantici nativi (es. TypeDB, Neo4j con estensione RDF).

3. Sperimentazione con Tecnologie per Knowledge Graph: Avviare attività di ricerca e sviluppo per esplorare l'integrazione della piattaforma con tecnologie avanzate per la gestione della conoscenza, come i database a grafo (es. Neo4j con estensione semantics per RDF) e i database concettuali (es. TypeDB). L'obiettivo è valutare architetture alternative per la rappresentazione e l'interrogazione di dati scientifici complessi e interconnessi.

Questa strategia evolutiva permetterà di aumentare progressivamente il livello di FAIRness del database, trasformandolo da un sistema ben strutturato a un vero e proprio knowledge graph interoperabile al servizio della ricerca italiana sui materiali per l'energia.

6 Conclusioni e prospettive

Al termine di questo percorso di analisi dettagliata, è possibile affermare con certezza che il database ENEA costituisce molto più di un semplice componente tecnico della piattaforma IEMAP; ne rappresenta la spina dorsale strategica, l'abilitatore fondamentale su cui poggiano tutte le ambizioni di condivisione, analisi e valorizzazione dei dati sui materiali per l'energia. Le modalità operative e le scelte architetturali adottate hanno permesso di raggiungere gli obiettivi primari del progetto, gettando al contempo delle solide fondamenta per le evoluzioni future.
Il lavoro svolto ha portato alla realizzazione di un'infrastruttura di dati centralizzata, robusta e sicura, che oggi serve efficacemente la comunità di ricerca del progetto. I principali risultati ottenuti possono essere così sintetizzati:

- Creazione di un Repository Unificato: È stato implementato con successo un database in grado di accogliere e strutturare dati scientifici eterogenei, integrando in un unico ecosistema sia i metadati di processi computazionali che i dati grezzi e aggregati di attività sperimentali.

- Processo di Ingestione Controllato e Affidabile: Sono stati definiti e resi operativi canali di ingestione multipli (interfaccia web, modulo Python, REST API), governati da un rigoroso processo di validazione multilivello che garantisce l'integrità e la coerenza formale di ogni dato archiviato.

- Architettura di Conservazione Scalabile e Sicura: L'adozione di una soluzione ibrida, che combina la flessibilità di MongoDB Enterprise per i metadati con la resilienza del filesystem distribuito Ceph per i file fisici, garantisce la scalabilità e la perennità del patrimonio informativo.

- Buon Livello di Conformità ai Principi FAIR: L'analisi della FAIRness ha confermato l'eccellenza del database negli ambiti della reperibilità (Findability) e dell'accessibilità (Accessibility), grazie all'uso di identificatori univoci e protocolli di accesso standard. Sebbene l'interoperabilità rappresenti un'area di miglioramento, la creazione di un'ontologia OWL formale costituisce un asset strategico di inestimabile valore per il futuro.

- Dimostrazione di Flessibilità: L'integrazione di casi d'uso specializzati, come l'archiviazione di dati sperimentali in collezioni Time Series per il processo di deposizione di perovskite, ha validato la capacità del database di adattarsi a esigenze specifiche e complesse.

In conclusione, il database ENEA nella sua forma attuale adempie pienamente alla sua missione di fornire un backbone di dati affidabile, organizzato e sicuro per la piattaforma IEMAP.

6.1 Prospettive e Roadmap Futura

Il lavoro svolto non rappresenta un punto di arrivo, ma una tappa fondamentale di un percorso evolutivo. Le analisi condotte e le esperienze maturate hanno permesso di delineare una chiara roadmap per il futuro, mirata a trasformare il database da un eccellente sistema di archiviazione a un vero e proprio knowledge graph al servizio della scienza dei materiali.
Le direttrici di sviluppo future includono:

Pieno Compimento della Visione FAIR attraverso la Semantica: L'evoluzione più strategica riguarderà il potenziamento dell'interoperabilità. La roadmap prevede di capitalizzare l'ontologia OWL già sviluppata attraverso le seguenti azioni:

- Pubblicazione e versionamento dell'ontologia su repository aperti come GitHub per favorirne l'adozione e l'evoluzione collaborativa.
- Mapping verso vocabolari standard internazionali (es. EMMO, ChEBI), per collegare i dati di IEMAP all'ecosistema globale dei dati scientifici.
- Sperimentazione di tecnologie semantiche native, come i database a grafo (es. TypeDB), per integrare l'ontologia direttamente nei meccanismi di interrogazione e validazione del database.

Rilascio e Potenziamento della Ricerca Intelligente: La funzionalità di query semantica, già sviluppata e testata con successo in ambiente di sviluppo, rappresenta la prossima grande innovazione per l'utente finale. Il piano prevede il rilascio in produzione della ricerca ibrida (basata su database vettoriale Qdrant) all'interno del modulo Python iemap-mi, per offrire agli utenti avanzati la capacità di interrogare i metadati in linguaggio naturale.

Evoluzione verso un'architettura completa di Retrieval-Augmented Generation (RAG). L'integrazione di Large Language Models (LLM) permetterà di passare da una semplice ricerca semantica a un'interazione conversazionale con i dati, dove l'utente potrà "dialogare" con il database per formulare analisi complesse e ottenere insight.

Generalizzazione dei Modelli per Dati Sperimentali: Partendo dall'esperienza positiva ma specifica del caso d'uso sulla perovskite, un'area di sviluppo futuro consisterà nel definire schemi e template di metadati più generalizzati per altre tipologie di processi sperimentali, al fine di accelerare e standardizzare l'onboarding di nuove e diverse fonti di dati di laboratorio.

Integrazione con Workflow Scientifici Automatizzati: Il modulo Python iemap-mi apre la strada a un'integrazione più profonda del database con piattaforme di gestione di workflow computazionali (come AiiDA). L'obiettivo è automatizzare il processo di conferimento dei dati e la cattura della provenienza direttamente alla fonte, riducendo l'onere manuale per i ricercatori e garantendo una tracciabilità completa e automatica del dato scientifico, dalla sua generazione alla sua pubblicazione sulla piattaforma.

Attraverso queste direttrici, il database ENEA si evolverà per diventare un asset dinamico e sempre più intelligente, in grado non solo di conservare, ma di connettere e valorizzare attivamente la conoscenza prodotta dalla comunità scientifica italiana nel strategico settore dei materiali per l'energia.

Articoli di approfondimento

Questa sezione include gli articoli seminali che hanno introdotto i principi FAIR e le rassegne che ne discutono l'applicazione e l'evoluzione.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3, 160018.
Perché è utile: È l'articolo fondamentale e più citato che ha introdotto e definito formalmente i principi FAIR. È il punto di partenza essenziale per comprendere la filosofia e gli obiettivi dietro l'acronimo.
DOI: 10.1038/sdata.2016.18

Collins, S., Genova, F., & Harrower, N. (2018). Turning FAIR into reality: Final report and action plan from the European Commission expert group on FAIR data.
Perché è utile: Questo report della Commissione Europea è un documento chiave che traduce i principi teorici in un piano d'azione concreto per l'implementazione su larga scala, in particolare nel contesto dell'European Open Science Cloud (EOSC).
DOI: 10.2777/1524

Jacobsen, A., de Miranda Azevedo, R., Juty, N., et al. (2020). FAIR Principles: Interpretations and Implementation Considerations. Data Intelligence, 2(1-2), 10-29.
Perché è utile: Un'eccellente analisi delle sfide pratiche e delle diverse interpretazioni che emergono quando si cerca di implementare i principi FAIR in progetti reali. Offre una visione critica e pragmatica.
DOI: 10.1162/dint_a_00024

Mons, B. (2020). The FAIR Principles as a pillar for the European Open Science Cloud. Library Hi Tech, 38(2), 295-301.
Perché è utile: Scritto da uno dei co-autori del paper originale, questo articolo contestualizza il ruolo cruciale della FAIRness come pilastro portante delle grandi infrastrutture di ricerca europee come l'EOSC.
DOI: 10.1108/LHT-12-2019-0245

I seguenti articoli si concentrano su come applicare concretamente i principi FAIR nell'architettura di database e piattaforme di dati, con esempi specifici.

Stocker, M., & Wyk, F. V. (2022). Ten simple rules for FAIR data and software. PLOS Computational Biology, 18(11), e1010653.
Perché è utile: Fornisce una guida pratica e accessibile in 10 regole per ricercatori e sviluppatori su come rendere FAIR non solo i dati ma anche il software, un aspetto cruciale per la riproducibilità.
DOI: 10.1371/journal.pcbi.1010653

Talirz, L., Kumbhar, S., Passarotti, M., et al. (2020). Materials Cloud, a platform for open and FAIR computational science. Scientific Data, 7, 299.
Perché è utile: Estremamente pertinente per il contesto di IEMAP, questo articolo descrive l'architettura di Materials Cloud, una piattaforma di successo per la condivisione di dati computazionali nel campo della scienza dei materiali, costruita interamente sui principi FAIR.
DOI: 10.1038/s41597-020-00637-5

Eguinoa, I., Bezdicek, M., Laporte, M. A., et al. (2022). The FAIR-agro platform: A FAIR approach to data management in agroecology. Computers and Electronics in Agriculture, 199, 107147.
Perché è utile: Presenta un caso di studio dettagliato sulla costruzione di una piattaforma di dati FAIR in un dominio scientifico specifico (agroecologia), offrendo spunti architetturali e operativi trasferibili ad altri contesti.
DOI: 10.1016/j.compag.2022.107147

Holub, P., Kozubek, M., & Kouřil, D. (2021). Data-retention and FAIR-data-management policies for life-science research departments. EMBO reports, 22(11), e53835.
Perché è utile: Si concentra su un aspetto spesso trascurato: la necessità di definire policy istituzionali chiare per la gestione e conservazione dei dati, che sono il prerequisito per un'effettiva implementazione della FAIRness.
DOI: 10.15252/embr.202153835

La seguente selezione affronta il principio dell'Interoperabilità, mostrando come ontologie e knowledge graph siano strumenti essenziali per creare dati semanticamente ricchi e comprensibili dalle macchine.

Cox, S. J., Atkinson, R., & Pouchard, L. C. (2021). Ten simple rules for making a vocabulary FAIR. PLOS Computational Biology, 17(6), e1009041.
Perché è utile: Una guida pratica e diretta su come creare e pubblicare vocabolari e ontologie che siano a loro volta FAIR, un passo fondamentale per garantire l'interoperabilità dei dati che li utilizzano.
DOI: 10.1371/journal.pcbi.1009041

Hogan, A., Blomqvist, E., Cochez, M., et al. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), 1-37.
Perché è utile: Una rassegna completa e autorevole sulla tecnologia dei Knowledge Graph. Sebbene non focalizzata solo su FAIR, spiega i fondamenti tecnologici (RDF, OWL, SPARQL) necessari per implementare la "I" e la "R" di FAIR al loro massimo potenziale.
DOI: 10.1145/3447772

Gray, A. J. G., Groth, P., & Peroni, S. (2021). An Introduction to the FAIR Principles and Research Data Management for Humanities Scholars. Journal of the Text Encoding Initiative, (14).
Perché è utile: Anche se orientato alle discipline umanistiche, questo articolo offre una delle spiegazioni più chiare e didattiche del ruolo delle ontologie e dei metadati strutturati per ottenere l'interoperabilità, con esempi molto comprensibili.
DOI: 10.4000/jtei.3663

Infine, una selezione di risorse web gestite da organizzazioni leader nel campo, utili per rimanere aggiornati e trovare guide pratiche.

GO FAIR Initiative Website

Perché è utile: È il sito ufficiale dell'iniziativa GO FAIR, il principale punto di riferimento per la comunità globale che lavora sull'implementazione dei principi. Contiene notizie, eventi, e documenti strategici.

URL: https://www.go-fair.org/

Australian Research Data Commons (ARDC) - "Making Data FAIR"

Perché è utile: L'ARDC è un'organizzazione leader a livello mondiale. La loro sezione sulla FAIRness è ricca di guide, webinar e checklist pratiche di altissima qualità, ideali per chi deve implementare i principi in un progetto.

URL: https://ardc.edu.au/resources/about-data/fair-data/

Articolo "FAIR is not a standard, it is a journey" (Erik Schultes, 2020) - FAIRplus Consortium

Perché è utile: Un post illuminante che chiarisce un concetto chiave: la FAIRness non è uno stato binario (si/no) da raggiungere, ma un processo di miglioramento continuo. Utile per gestire le aspettative e pianificare una roadmap realistica.

URL: https://fairplus-project.eu/news/fair-not-standard-it-journey

DTL (Dutch Techcentre for Life Sciences) - "How to FAIR"

Perché è utile: Fornisce una serie di tutorial e strumenti pratici ("How-tos") per affrontare specifici aspetti della FAIRness, dalla scelta dei formati di file alla creazione di metadati. Molto orientato alla pratica.

URL: https://www.health-ri.nl/services/fair-data-stewardsh