Federation-Registry: la nuova versione del Configuration Management Database per la federazione dinamica delle cloud

G. Savarese*, M. Antonacci, INFN-BA, Bari (BA), Italy,
L. Giommi, INFN-CNAF, Bologna (BO), Italy

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

LO STATO DELL'ARTE

L'**INDIGO-PaaS Orchestrator**, per eseguire una procedura di deployment, sfrutta diversi micro-servizi. Alcuni di questi, non sono più mantenuti o non soddisfano più i nostri requisiti:

Configuration Management Database (CMDB)

- REST API scritta in **Java8** per gestire le informazioni dei provider federati e i loro servizi.
- Utilizza **Apache CouchDB** (database non relazionale).

Non sviluppato da noi e non più mantenuto.

Apache CouchDB non soddisfa più le nostre necessità.

Cloud Info Provider (CIP)

- Script **python2.7** che popola il CMDB a partire da files YAML.

Basato su python2.7.

Fortemente legato alla struttura dei dati del CMDB.

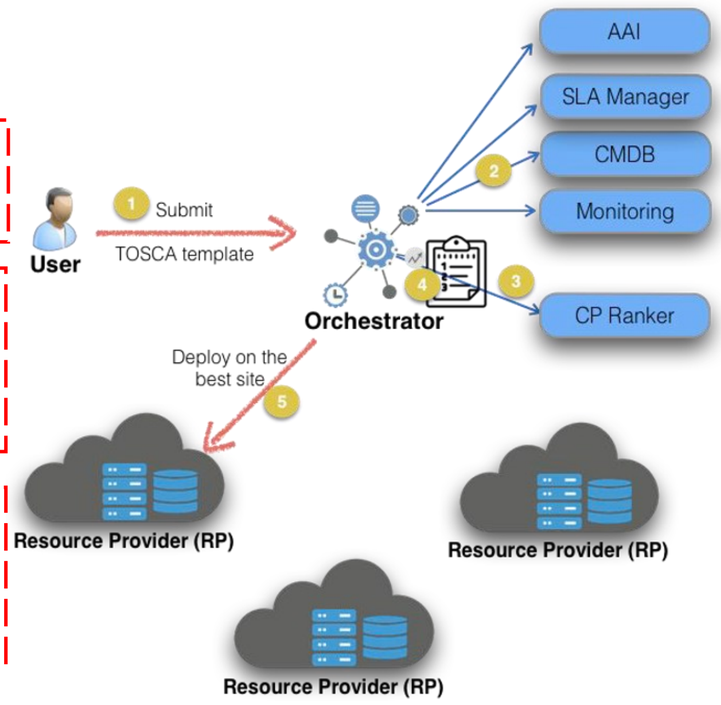
YAML file migliorabili.

Service Level Agreement Tool (SLAT)

- REST API scritta in **Python3**. Utilizza **Flask** e gestisce le quote sull'utilizzo delle risorse dei provider associate ai gruppi di utenti.
- Utilizza **MySQL** (database relazionale).

Fortemente legato alla struttura dei dati del CMDB.

Duplicazione delle informazioni contenute nel CMDB.



LA SOLUZIONE

Sostituire **CMDB** e **SLAT** con un nuovo servizio: il **Federation-Registry**. Il nuovo servizio punta a: ridurre la dispersione e la duplicazione delle informazioni, rimpiazzare i componenti non più mantenuti con nuove tecnologie, riorganizzare i dati in maniera più funzionale per l'Orchestratore e con la predisposizione per l'integrazione di nuove tipologie di provider.

Rimpiazzare **CIP** con uno il **Federation-Registry-Feeder**. Questo script usa dei file YAML per definire la lista dei provider da federare, recupera direttamente dai provider federati la configurazione dei servizi ed è autorizzato ad eseguire operazioni di scrittura sul Federation-Registry.

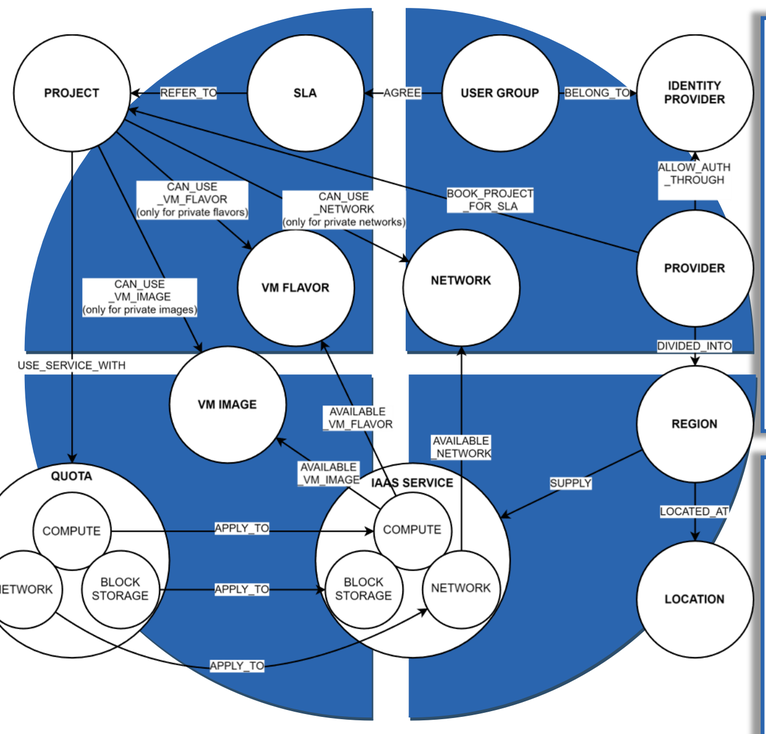
DESIGN DEL FEDERATION-REGISTRY

Python3

- La frequenza delle richieste fatte all'Orchestratore e il tempo di risposta sono **trascurabili** rispetto il tempo impiegato dalla procedura di deployment.
- Ogni richiesta REST è incapsulata in una **singola transazione** col database al fine di mantenere *l'atomicità delle operazioni* e la *consistenza dei dati*.

Neo4j

- Utilizza un database a grafi. Vantaggi dei DB relationali e non-relazionali
- Scala efficientemente quando si devono gestire enormi dataset.
- Usa il linguaggio **Cypher** per eseguire le query



FastAPI

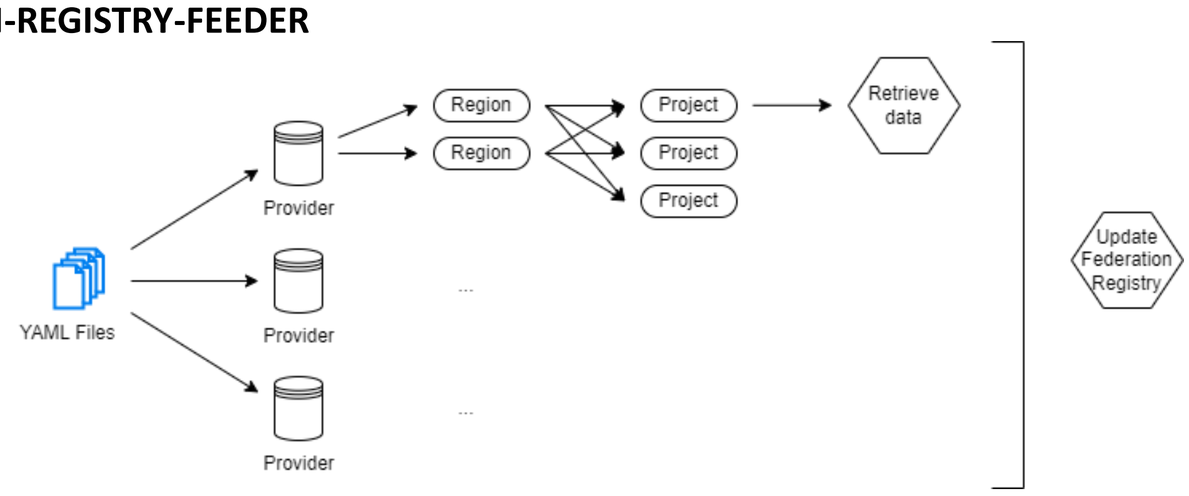
- **Prestazioni al pari di NodeJS e Go**
- **Meno codice e meno duplicazione.**
- Implementa gli **open standards per le API.**
- Genera automaticamente la documentazione online delle API con la possibilità di eseguire query.
- Supporta *pydantic* per validare i dati in ingresso.

Autenticazione e Autorizzazione

- La libreria python **flaat** supporta l'autenticazione tramite qualsiasi identity provider che implementi il protocollo OIDC.
- Chiunque può leggere i dati, solo utenti autorizzati possono eseguire operazioni di scrittura.

DESIGN DEL FEDERATION-REGISTRY-FEEDER

- Script basato su python3 eseguito **periodicamente.**
- Basato su file YAML. Struttura espandibili per accettare nuove tipologie di provider.
- Sfrutta **OIDC-Agent** per recuperare gli access token **per ispezionare gli endpoint dei provider da federare** e per eseguire operazioni di **lettura e scrittura sul Federation Registry.**




```yaml
trusted_idps:
  - issuer: https://iam.cloud.infn.it/
    group_claim: groups
    user_groups:
      - name: test
        slas:
          - doc_uuid: edfda059a23c439f8ffc06edd484e1a0
            start_date: 2023-09-10
            end_date: 2024-04-25

openstack:
  - name: recas-ba
    status: active
    is_public: false
    support_emails:
      - admin-test@ba.infn.it
    regions:
      - name: RegionOne
        location:
          site: INFN Bari
          country: Italy
    auth_url: https://keystone.recas.ba.infn.it:443
    identity_providers:
      - name: infn-cloud
        protocol: openid
        endpoint: https://iam.cloud.infn.it/
    projects:
      - id: a8b324a0f4f349a28e98e4e78b11bacc
        sla: edfda059a23c439f8ffc06edd484e1a0
```

Procedura

Legge i file YAML e *per **ciascun provider*** recupera, tramite OIDC-agent, un access token valido associato all'utente *ops*.

*Per **ciascun provider**,* per ogni possibile combinazione ***progetto-regione***, si connette al provider e recupera le risorse disponibili per quel progetto (flavors, immagini, reti, quote e altro).

Unifica le informazioni recuperate e le compara con le informazioni attualmente contenute nel Federation-Registry.

Modifica tramite API i dati contenuti nel Federation-Registry in modo da mantenerlo aggiornato con le configurazioni attuali dei provider.

FUNZIONALITÀ E STRUMENTI COMUNI

- Test basati su **pytest**, **pytest-mock**, **pytest-cases** e **pytest-coverage**
- Valutazione del codice con **SonarCloud**
- **Github actions**
- Immagini **docker di produzione e sviluppo (devcontainer)**
- Gestione delle dipendenze con **poetry**.