UNIVERSITÀ
degli STUDI
di CATANIA

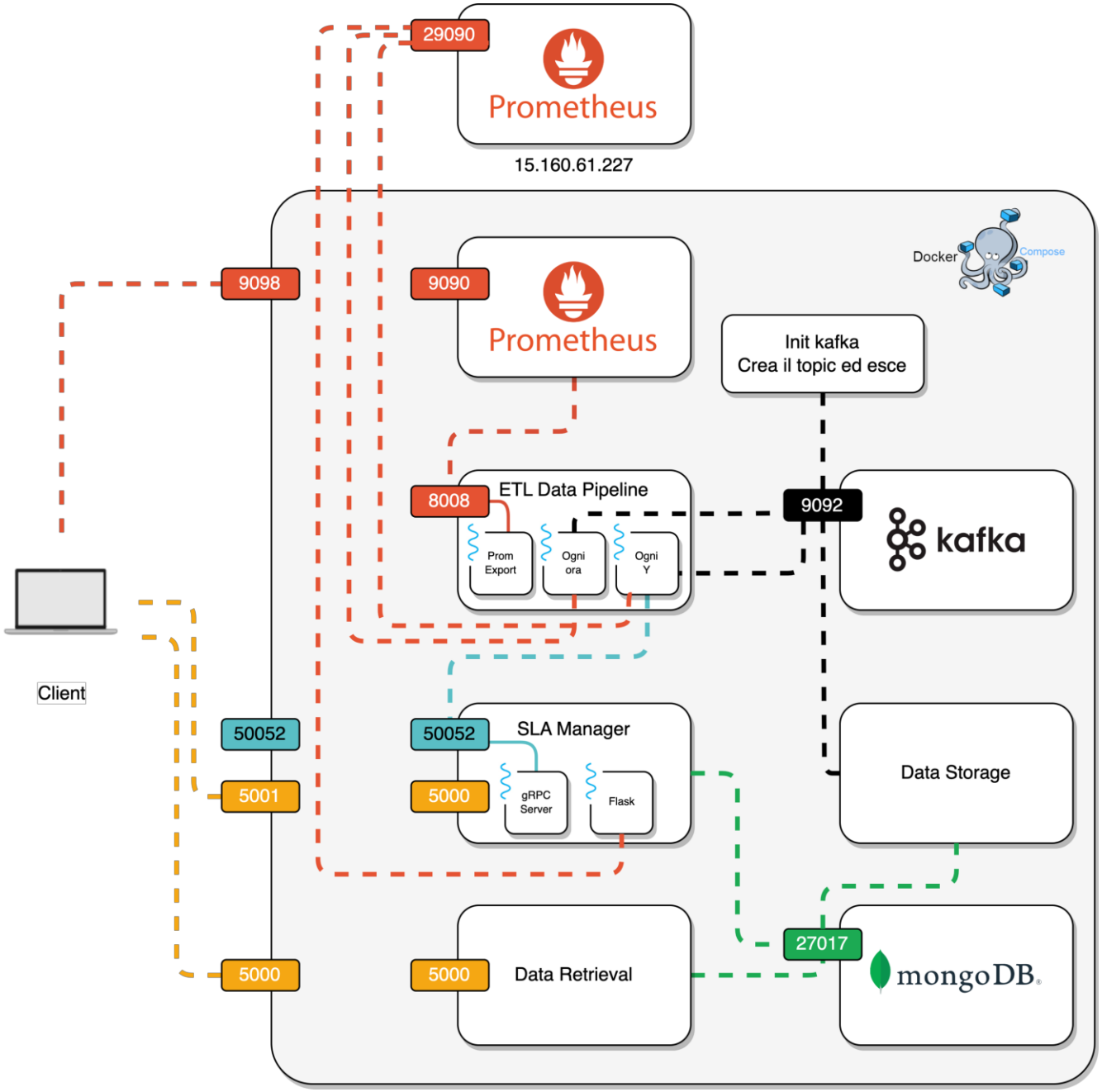
PROGETTO IN ITINERE

Corso di Distribuited System and Big Data

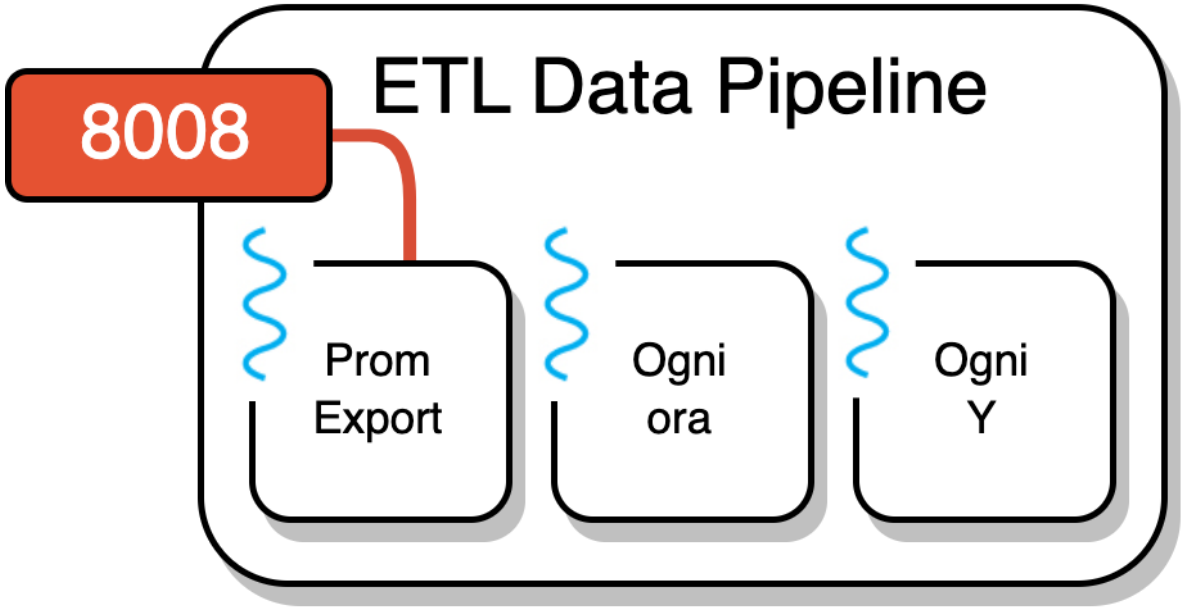
A.A. 2022-2023

Alessio Pirri 1000040385 Mattia Pirri 1000042320

ARCHITETTURA

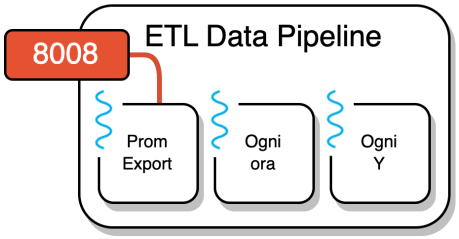


ETL DATA PIPELINE



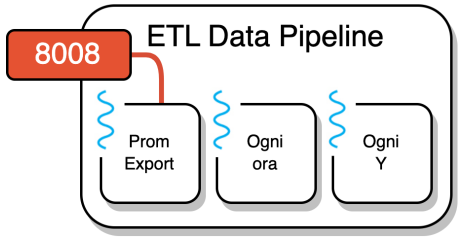
1. Calcolo dei metadati
2. Calcolo dei valori di aggregazione e predizione
3. Exporter Prometheus

ETL DATA PIPELINE



1. Calcolo dei metadati
   - Schedulato ogni ora
   - Stazionarietà tramite ADF (p-value < 0.05)
   - Autocorrelazione i tramite la funzione ACF di StatsModels
   - Stagionalità tramite la trasformata di Fourier
     - Affinamento del risultato tramite minimizzazione dell'errore residuo della **SeasonalDecompose** distinguendo il modello additivo da quello moltiplicativo
2. Calcolo dei valori di aggregazione e predizione
3. Exporter Prometheus

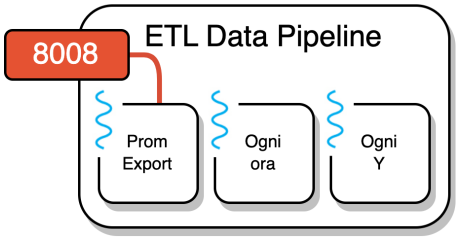
ETL DATA PIPELINE



1. Calcolo dei metadati
2. Calcolo dei valori di aggregazione e predizione

   Schedulato ogni 2 min ottiene SLA set tramite **gRPC** e calcola i valori massimo, minimo, medio e deviazione standard per ogni metrica del Prometheus. Mentre per ogni metrica del set calcola la predizione tramite **ExponentialSmoothing** utilizzando i periodi (calcolati con il modello additivo e con il modello moltiplicativo) e sceglie quella che **minimizza l'errore**. Per ottimizzare i tempi di esecuzione viene fatto il **resampling** ogni 2 min se il periodo è maggiore di 720 min e ogni 5 min se il periodo è maggiore di 1440. Per ottimizzare ulteriormente, nel caso in cui la metrica fosse costante, viene creata manualmente la predizione con il valore osservato
3. Exporter Prometheus

ETL DATA PIPELINE



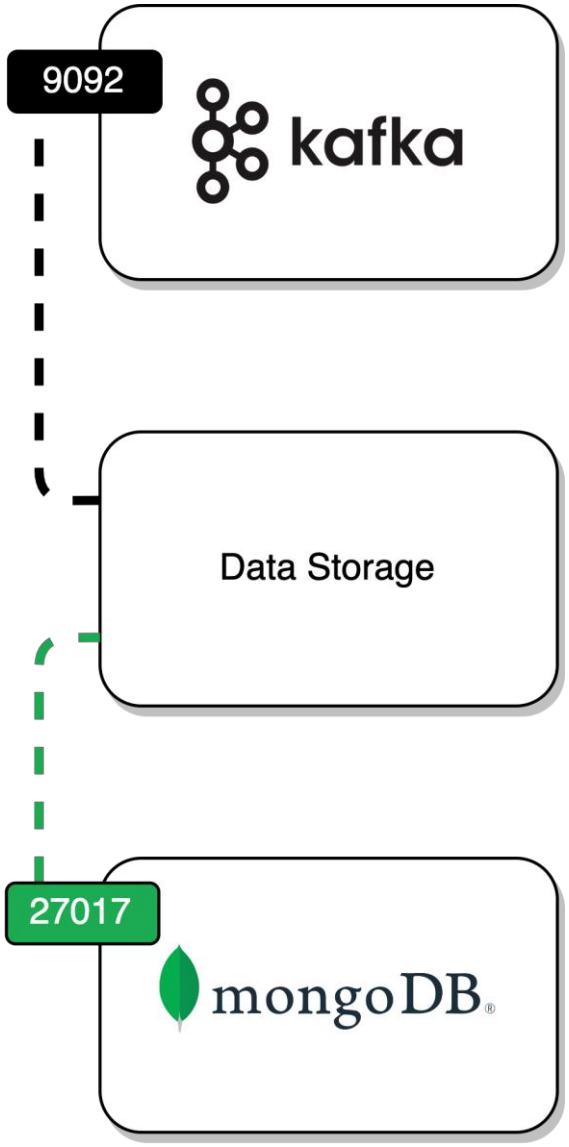
1. Calcolo dei metadati
2. Calcolo dei valori di aggregazione e predizione
3. Exporter Prometheus

   Per monitorare i tempi di calcolo delle varie funzionalità. I dati sono disponibili tramite HTTP sulla porta *8008* nel formato Prometheus. In particolare espone le seguenti metriche:

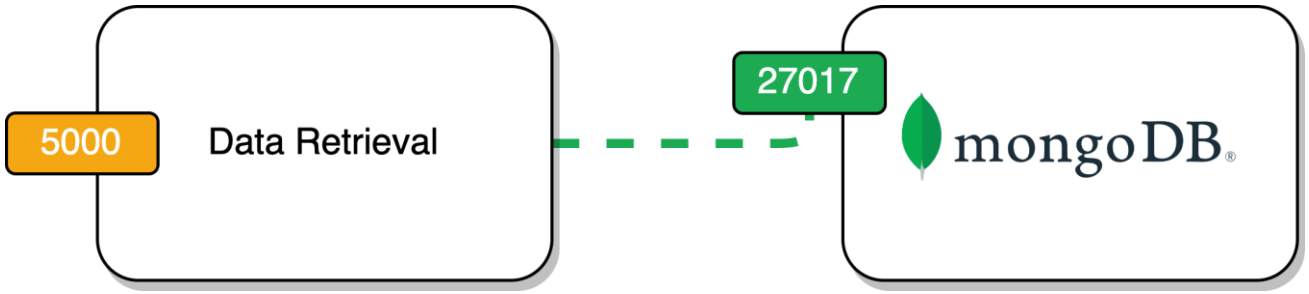
   - **etl_executiontime_<x>h**: con x ∈ [1, 2, 12]. Per ogni metrica viene calcolato il tempo di esecuzione per calcolare i dati di aggregazione nelle ultime x ore;
   - **etl_executiontime_metadata**: monitora il tempo di esecuzione speso per calcolare i metadati;
   - **etl_executiontime_prediction**: monitora il tempo impiegato per: l'allenamento dei modello predittivi, la predizione e il confronto degli errori per scegliere il modello più adatto

DATA STORAGE

Legge i messaggi dal topic «prometheusdata» e per ogni messaggio ricevuto effettua un upsert del documento nel database MongoDB.
Il commit automatico della lettura su Kafka è stato disabilitato e viene effettuato manualmente subito dopo la scrittura nel DB. Nel caso in cui il database non fosse disponibile si ritenta in seguito l'operazione.



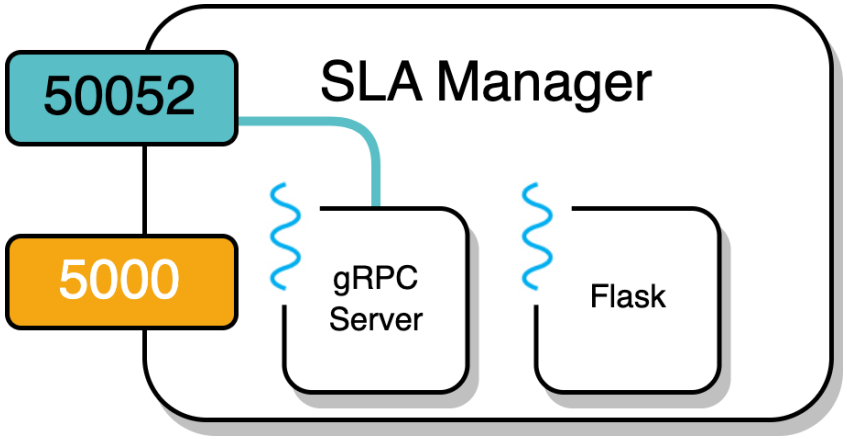
DATA RETRIEVAL



Microservizio che esegue un'applicativo Flask che offre un'interfaccia REST sulla porta 5000. Si occupa di recuperare le informazioni presenti nel database MongoDB e mostrarle all'utente in modo strutturato.

- /metrics
- /metrics/{id}/metadata
- /metrics/{id}/values
- /metrics/{id}/prediction

SLA MANAGER



Microservizio che in due thread separati esegue un'applicazione Flask esponendo le API descritte nelle slide successive e un server gRPC per la comunicazione con l'ETL Data Pipeline. In fase di definizione dell'SLA set è necessario inserire una lista di 5 metriche per ognuna delle quali si deve definire un range di valori ammessi specificando massimo e minimo e il tempo, espresso in minuti, dopo il quale se la metrica esce dal range dei valori ammessi scatta l'allarme

SLA MANAGER

- /slaset [GET]

  Ritorna l'SLA set

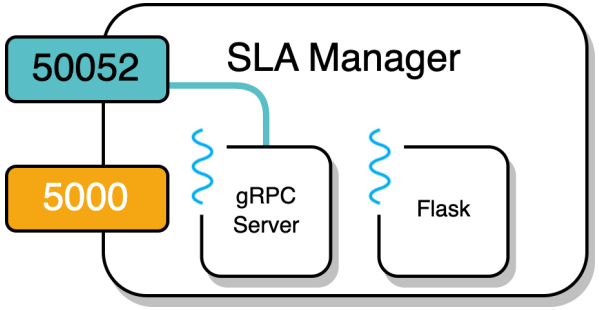
- /slaset [PUT]

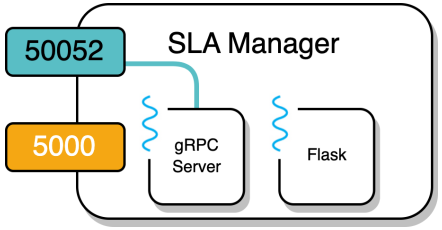
  Aggiorna o crea l'SLA set

- /status

  Ritorna lo stato dell'SLA.
  Per ogni metrica ritorna lo stato che può essere inactive, pending o firing e un campo description che sarà rispettivamente per i tre casi precedenti:
  - Actual value [{}] in ({}, {})
  - Service level agreement violated for {}/{} min({%}), last three values [{},{},{}] not in ({}, {})
  - Service level agreement violated for more than {} min, last three values {}, allowed range: ({}, {})

  Ritorna inoltre gli ultimi tre campioni della serie.





- /violations

  Per ogni metrica appartenente al set
  ritorna il numero di violazioni e il tempo totale di violazione in 1, 3 e 12 ore e la lista delle violazioni (instanti di inizio e di fine, durata, valori min, max e medio)

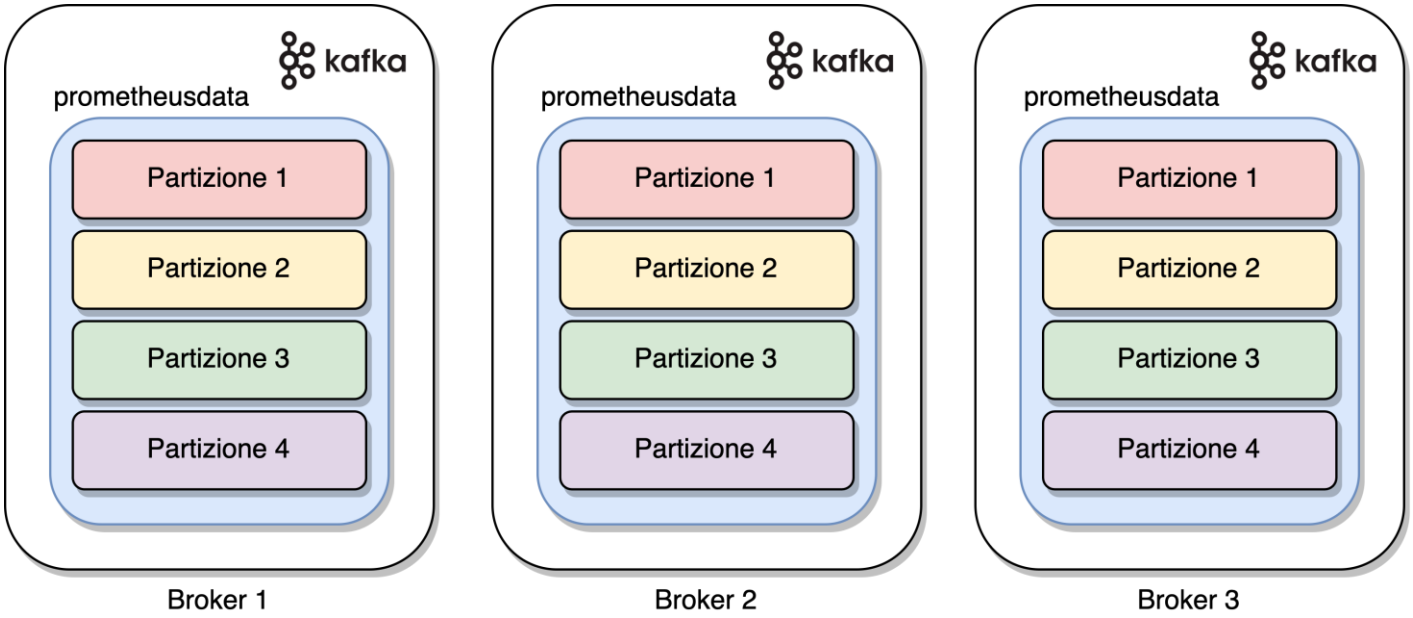
- /future-violations

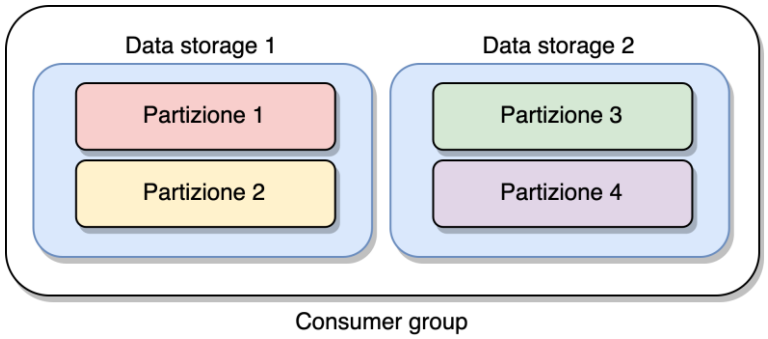
  Ritorna la lista delle possibili violazioni nei successivi 10 minuti, per ogni metrica appartenente al set.
  Per ogni violazione viene riportato: il timestamp di inizio e fine violazione, la durata, i valori max, avg e min e una descrizione che puo essere:
  - Was already firing, continuing firing
  - Was pending, will fire
  - Will fire
  - Will be pending

  Viene ritornato inoltre l'errore effettuato durante la predizione

SLA MANAGER

REPLICAZIONE KAFKA





Nel Deployment **Kubernetes** abbiamo deciso di realizzare un cluster composto da 3 broker.

Il topic kafka è stato creato con quattro partizioni e replication factor pari a 3, così da ottenere high availability e scalabilità.