

# Cloud Computing Technologies **Project**

Luca Milanesi 55053A  
luca.milanesi1@studenti.unimi.it

A.Y. 2024-2025

# Indice

<b>1</b>	<b>Descrizione</b>	<b>3</b>
<b>2</b>	<b>Message Broker</b>	<b>3</b>
<b>3</b>	<b>Microservizi</b>	<b>4</b>
3.1	Simulatore di Eventi: Producer . . . . .	4
3.2	Analisi Dati: Consumer . . . . .	4
<b>4</b>	<b>Implementazione dei requisiti non funzionali</b>	<b>5</b>
4.1	Fault Tolerance & High Availability . . . . .	5
4.2	Scalabilità . . . . .	5
4.3	Sicurezza . . . . .	5

# 1 Descrizione

Questo progetto implementa un semplice sistema di analisi dati per un e-commerce. L'architettura del sistema si basa su un modello a microservizi, composto da un produttore (simulatore) di eventi e un consumatore di eventi per l'analisi, che comunicano, stando al pattern producer-consumer, tramite un cluster Kafka.

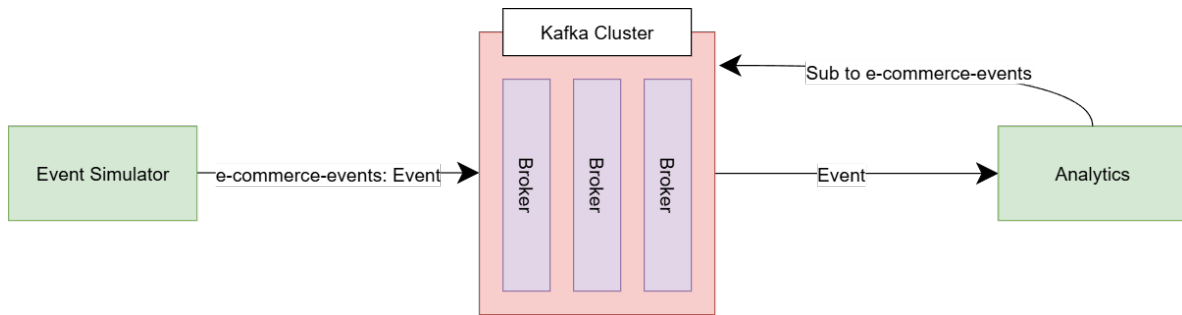


Figura 1: Architettura del progetto.

Come illustrato nell'architettura, il microservizio **Event Simulator** funge da produttore e simula una serie di eventi utente tipici di una piattaforma e-commerce, inviandoli a dei topic Kafka. Il microservizio **Analytics** agisce come consumatore, leggendo gli eventi dal topic degli eventi utente per elaborare e salvare semplici statistiche di marketing su un database dedicato.

## 2 Message Broker

Per garantire fault tolerance e HA, si implementerà un cluster Kafka composto da più broker con replica delle partizioni dei topics. Avere un cluster con più broker e partizioni replicate assicura che il sistema possa continuare a funzionare anche se uno o più nodi non sono disponibili. Se un broker fallisce, i consumatori possono continuare a leggere i dati da un broker replicato, garantendo la disponibilità continua del servizio.

Inoltre, per soddisfare i requisiti di sicurezza, verranno implementate crittografia e autenticazione. La crittografia proteggerà i dati in transito tra i microservizi e il cluster, mentre l'autenticazione assicurerà che solo i servizi autorizzati possano connettersi al cluster.

## 3 Microservizi

### 3.1 Simulatore di Eventi: Producer

Questo microservizio simulerà il comportamento degli utenti su un classico sito di e-commerce, generando vari eventi e inviandoli al cluster Kafka sul topic apposito. Ad ogni evento saranno associate delle informazioni generali come session e user ID, posizione e timestamp. Gli eventi specifici che verranno prodotti sono:

- Inizio di una sessione utente, con l’user-agent.
- Fine di una sessione, con la durata complessiva.
- Visualizzazione di una specifica categoria di prodotti.
- Visualizzazione di un prodotto specifico, con le relative informazioni.
- Aggiunta di un prodotto al carrello, con relativa quantità.
- Rimozione di un prodotto dal carrello, con relativa quantità.
- Ordine di acquisto, con dettagli su prodotti, quantità, luogo di consegna, sconti e metodi di pagamento.

Gli eventi verranno inviati in modo continuo, simulando un flusso di dati realistico, simile a quello dei principali siti di e-commerce reali.

Per semplicità, i dati degli eventi saranno dei dati di mock, ovvero dati pregenerati e simulati.

### 3.2 Analisi Dati: Consumer

Questo microservizio si iscriverà al topic degli eventi utente nel cluster Kafka e li consumerà per calcolare e salvare in maniera persistente semplici statistiche di marketing in un database apposito, che sarà dedicato unicamente a questo singolo microservizio. Le statistiche raccolte, in uno scenario realistico, potrebbero essere usate per comprendere il comportamento generale del mercato e l’efficacia delle strategie di vendita. Le metriche calcolate potrebbero supportare l’analisi dei seguenti aspetti:

- Numero totale di sessioni attive/terminate e la loro durata media.
- Categorie di interesse, una statistica basata su metriche come visualizzazione, aggiunta/rimozione al carrello e acquisto di un prodotto di una certa categoria, assegnando un peso ad ogni metrica.

- Prodotti di interesse, statistica simile alla precedente ma per i singoli prodotti.
- Prodotti di tendenza, i più acquistati.
- Tasso di conversione, calcolato come

$$\frac{\textit{acquisti}}{\textit{visualizzazioni}}$$

per misurare l'efficacia delle singole schede prodotto.

- Statistiche di attività per area geografica.
- Statistiche economiche quali: ricavi, numero di ordini completati e metodo di pagamento più usato.

## 4 Implementazione dei requisiti non funzionali

### 4.1 Fault Tolerance & High Availability

Il cluster Kafka sarà configurato con almeno tre broker, per avere il minimo sindacale di risorse per avere Fault Tolerance e HA. Inoltre, le partizioni dei topic verranno replicate su più broker, garantendo che i dati non vengano persi e che il servizio rimanga disponibile anche in caso di guasto di un broker.

Anche i microservizi producer e consumer verranno replicati, creando più istanze di ciascuno. Se un'istanza di un microservizio dovesse fallire, le altre istanze continueranno a funzionare, assicurando i requisiti richiesti.

### 4.2 Scalabilità

A livello di message broker, il cluster Kafka può essere scalato orizzontalmente aggiungendo nuovi broker, aumentando così capacità di archiviazione e trasmissione. A livello di microservizi, sia il producer che il consumer possono essere istanziati in più copie per aumentare la capacità di produzione ed elaborazione degli eventi, permettendo al sistema di gestire anche carichi di lavoro pesanti.

### 4.3 Sicurezza

Per garantire sicurezza, verranno implementate crittografia, autenticazione e autorizzazione nel cluster Kafka.

La crittografia del canale sarà abilitata utilizzando SSL/TLS, proteggendo i dati in transito sia tra i microservizi e i broker, sia tra i broker stessi.

L'autenticazione verrà gestita con SASL, che richiederà ai microservizi esterni di fornire credenziali valide per connettersi ai broker, impedendo gli accessi non autorizzati.

Infine, l'autorizzazione tramite ACL permetterà di controllare e gestire le operazioni consentite a ciascun utente sul cluster: ad esempio, il microservizio producer potrà scrivere sul topic di interesse, mentre il microservizio consumer avrà il solo permesso di leggerlo.