

***RELAZIONE HOMEWORK#2***  
***DISTRIBUTED SYSTEMS AND BIG DATA***

**Studenti:**

Nunzio Fornitto: 1000002901

Angelo Frasca : 1000067615

**Professori:**

Antonella Di Stefano

Giovanni Morana

**Tutor:**

Massimo Gollo

# Abstract

In questo HW2, sono stati aggiunti vari componenti e cambiati i meccanismi di lettura e scrittura dei componenti che accedono al db, usando il pattern CQRS. Sono state inserite le soglie high value e low value per l'azione di interesse dell'utente, in modo da inviare un messaggio di notifica all'utente nel caso una delle due soglie sia superiore o inferiore rispetto al valore del ticker di interesse. Le soglie possono essere inserite dall'utente sia in fase di registrazione che di aggiornamento.

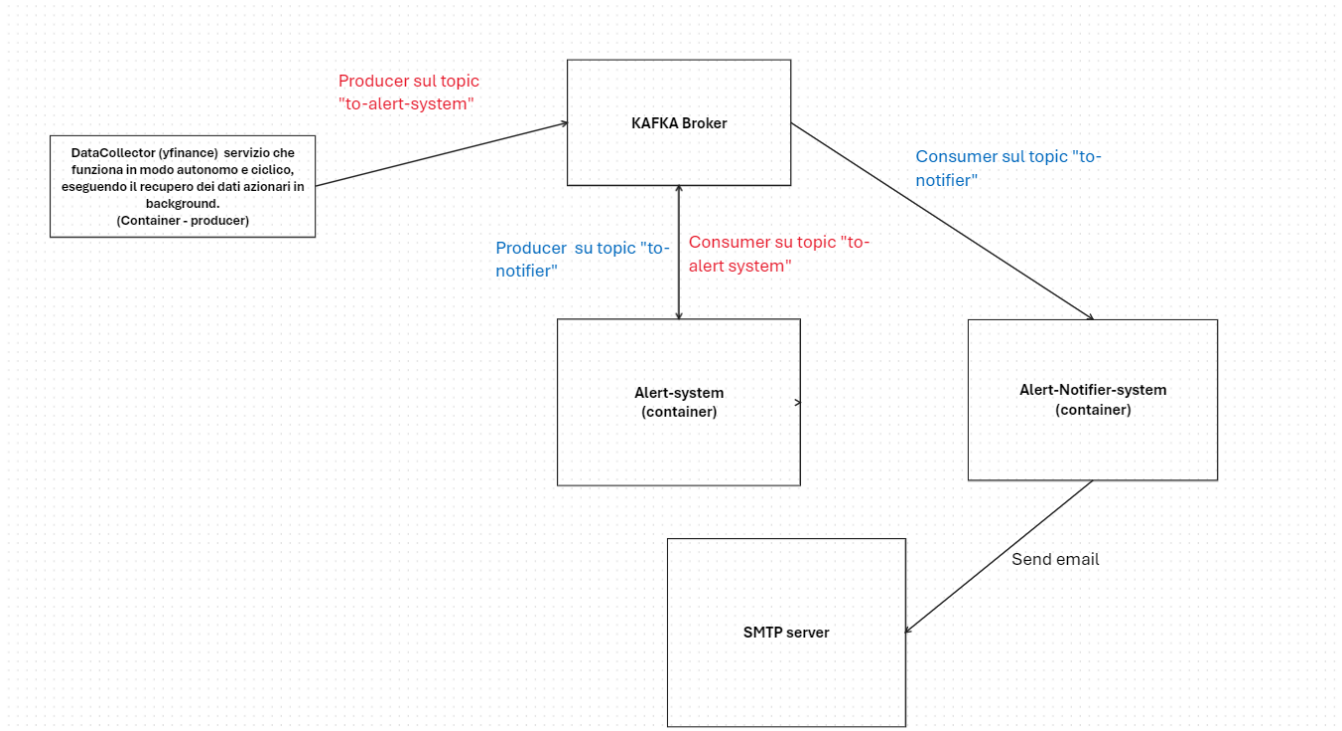
In particolare rispetto all'hw1, in aggiunta è stato implementato un sistema di notifiche, usando un broker kafka, Il sistema delle notifiche che utilizza il broker, è costituito da tre componenti:

**Datacollector:** (vecchio componente, aggiornando con le scritture al db tramite il cqrs, ed è stato aggiunto il meccanismo per la pubblicazione di un messaggio sul topic "to-alert-system").

**Alert-system:** è sia un consumer sottoscritto al topic "To-Alert-System" che un producer che pubblica sul topic "To-Notify" : appena avviene il poll del messaggio viene effettuata una query di lettura nel database MySQL che recupera il valore più recente del ticker per ogni utente (timestamp), se tale valore supera la soglia high\_value o è inferiore alla soglia low\_value, avviene la pubblicazione di un messaggio sul topic "to-Notify"; il messaggio è costituito da questi campi : email, ticker e condizione di superamento.

**Alert\_System\_Notify** è un consumer del topic "to-Notify", non appena effettua il poll di un messaggio impacchetta una mail che prontamente invia ad un server SMTP preconfigurato

# Diagramma architetturale dei nuovi componenti



# Diagramma di interazioni dei nuovi componenti

## Sistema Notifiche

1. Il DataCollector è un producer che a seguito di ogni aggiornamento del DB (ogni 200 sec) pubblica un msg sul topic "To-Alert-System" per notificare gli altri consumer del topic.

2. Il componente Alert\_System è sia un consumer sottoscritto al topic "To-Alert-System" e un producer che pubblica sul topic "To-Notify" : appena avviene il poll del msg viene effettuata una query di lettura nel database MySQL che recupera il valore più recente del ticker per ogni utente (timestamp), se tale valore supera la soglia high\_value o è inferiore alla soglia low\_value, avviene la pubblicazione di un msg sul topic "to-Notify"; il msg è costituito da questi campi : email, ticker e condizione di superamento;

3. Il componente Alert\_System\_Notify è un consumer del topic "to-Notify", non appena effettua il poll di un msg impacchetta una mail che prontamente invia ad un server SMTP preconfigurato.

