

Data Retention Hub

- [API](#)
- [How it works](#)
- [Architettura](#)
- [Integrazione](#)

API

Dev

<https://data-retention-write-dev.agyo.io/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>

Test

<https://data-retention-write-test.agyo.io/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>

Prod

<https://data-retention-write.agyo.io/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>

How it works

Lo scopo del servizio **Data Retention Hub** è quello gestire l'archiviazione e l'eliminazione di dati tramite un processo di scheduling configurabile tramite variabili d'ambiente.

Il flusso è il seguente.

Viene chiamata la *POST /api/v1/retentions*, la quale crea un record sulla tabella *archiving_request* contenente item e servizio.

Successivamente, un job schedulato recupera dalla tabella *archiving_request* i record da processare per archiviare i dati e crea un identico record in una seconda tabella chiamata *deletion_request*.

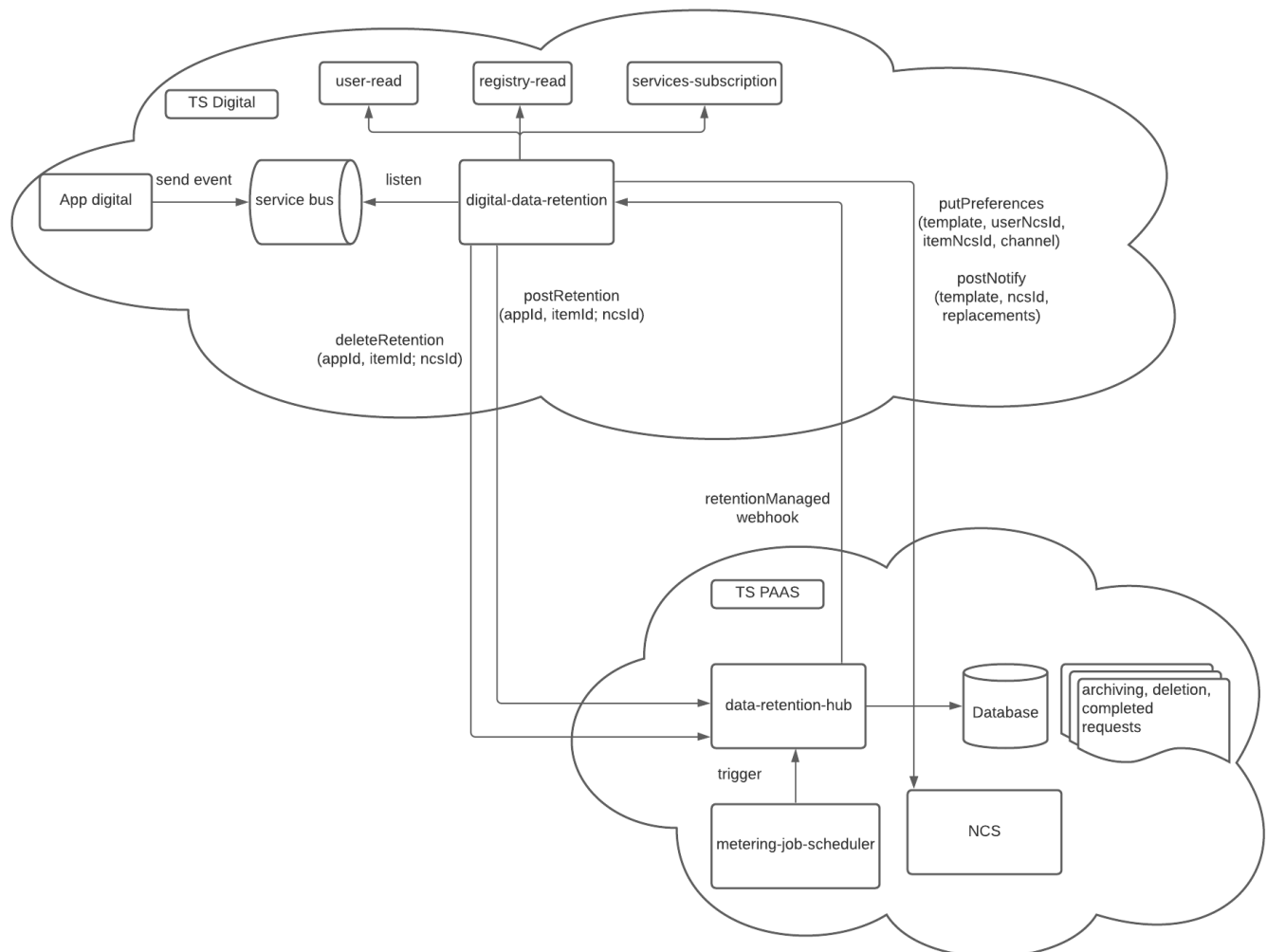
Un secondo job, recupera dalla tabella *deletion_request* i record da processare per eliminare i dati e crea un identico record in una terza tabella chiamata *completed_request*.

I cron per gestire i job sono configurabili tramite variabili d'ambiente.

Esiste infine un'api *DELETE /api/v1/retentions* per cancellare i record dalla *archiving_request* e *deletion_request*.

Chi chiama il servizio attualmente è il *Metering*, invocando la *POST* in seguito all'attivazione del readonly per un determinato servizio/cliente e la **DELETE** in seguito all'attivazione di un servizio (per un determinato servizio/cliente).

Architettura



Servizi coinvolti:

1. Digital-data-retention

E' un listener che legge dal service bus i messaggi relativi agli eventi di disattivazione o di messa in read only di un servizio ed esegue una richiesta di archiviazione dei dati. Dopo che l'archiviazione dati è stata effettuata, questo servizio si occupa anche di inviare le varie notifiche agli utenti

2. Data-retention-hub

E' il cuore del sistema di data retention. Gestisce l'archiviazione dei dati e la cancellazione dei dati e notifica il digital-data-retention quando un'operazione è stata effettuata.

3. Metering-job-scheduler

Servizio che gestisce il cron per lanciare le varie operazioni che data-retention-hub deve

svolgere.

Integrazione

Una app per potersi integrare con il sistema di data retention deve mettere a disposizione del servizio data-retention-hub le seguenti API:

- una POST per lanciare il job di archiviazione dati, che ritorna un 202
- una POST per lanciare il job di cancellazione dati, che ritorna un 202
- una POST per lanciare il job di rollback (es. il cliente riacquista la licenza), che ritorna un 202

Le API devono essere asincrone, in modo tale che data-retention-hub non rimanga appeso

L'autenticazione davanti a queste API è a carico dell'applicazione che si vuole integrare, deve essere predisposto un token per poter richiamare le API da parte di data-retention-hub. In caso di una app che usa il sistema di autenticazione di TS Digital, va benissimo un token generato da TSDigital.

La struttura degli header e body delle request dovrà essere identica per tutte 3 le API.

Esempio di cURL

```
curl --location --request POST 'http://localhost:8080/api/archive' \  
--header 'Content-Type: application/json' \  
--header 'X-App-Name: POSTMAN' \  
--header 'X-App-Version: 1.0.0' \  
--header 'X-Correlation-Id: e97120a7-b65e-4dac-85e7-311b0aa1b5fc' \  
--header 'X-Request-Id: 008f7273-8f01-48f7-b9fc-95422de6d77f' \  
--header 'Authorization: Bearer <TOKEN>' \  
--data-raw '{"jobData":{"itemId": "efd87ca0-b442-4d8a-8fa9-16afaea05a04"}}'
```