Moviesquik

Piattaforma per la condivisione e lo streaming di contenuti multimediali (movies, TV shows, ecc) in modalità on demand similmente alle piattaforme quali Netflix e Prime Video

Web project [main features] documentation for the “Web Computing” course

Agostino Rizzo - 176179

Funzionalità principali

- Gestione piani e abbonamenti
- Gestione streaming video attraverso l’implementazione di una CDN
- Raccolta statistiche sui contenuti
- Tracciamento delle preferenze e delle abitudini per la personalizzazione dei contenuti
- Gestione amministrazione, uploading dei contenuti, tracciamento e reporting del traffico (consultazione Google Analytics), configurazione e monitoraggio di una CDN (Context Delivery Network)
- Gestione e condivisione di watchlists
- Creazione di sale cinema virtuali (movie parties, gruppi) per la visualizzazione condivisa (multicasting) dei contenuti
- Gestione comunicazioni e pubblicazioni (chats, commenti, recensioni, ecc)

Implementazione CDN (Context Delivery Network)

Lo streaming dei contenuti è basato sull'implementazione di una CDN costituita da un insieme di Streaming Servers intesi per essere distribuiti su un'area geografica. E' possibile utilizzare un qualsiasi sistema (PC, Raspberry, ecc) opportunamente configurato e registrato nel database. Gli scripts di configurazione e funzionamento di un server sono inclusi nel progetto.

Ogni server è collocato in una posizione geografica. Una volta configurato, il server è in grado di consultare periodicamente il Web Server per aggiornare il suo stato e la sua disponibilità.

Ogni server è progettato per inviare periodicamente al Web Server alcuni parametri come lo stato di utilizzo espresso in requests/second. Tali informazioni saranno visibili in tempo reale attraverso opportuni grafici mediante l'accesso all'area Business dell'applicazione riservata ad accounts di amministrazione.

L'insieme dei servers costituisce una rete **peer-to-peer**. Lo streaming dei contenuti avviene attraverso la consultazione di uno o più Streaming Servers.

Consultazione Streaming Server (CDN manifest)

Ogni Streaming Server appartenente alla CDN è progettato per inviare periodicamente al Web Server alcune informazione sulla sua disponibilità e sullo stato di utilizzo (overhead).

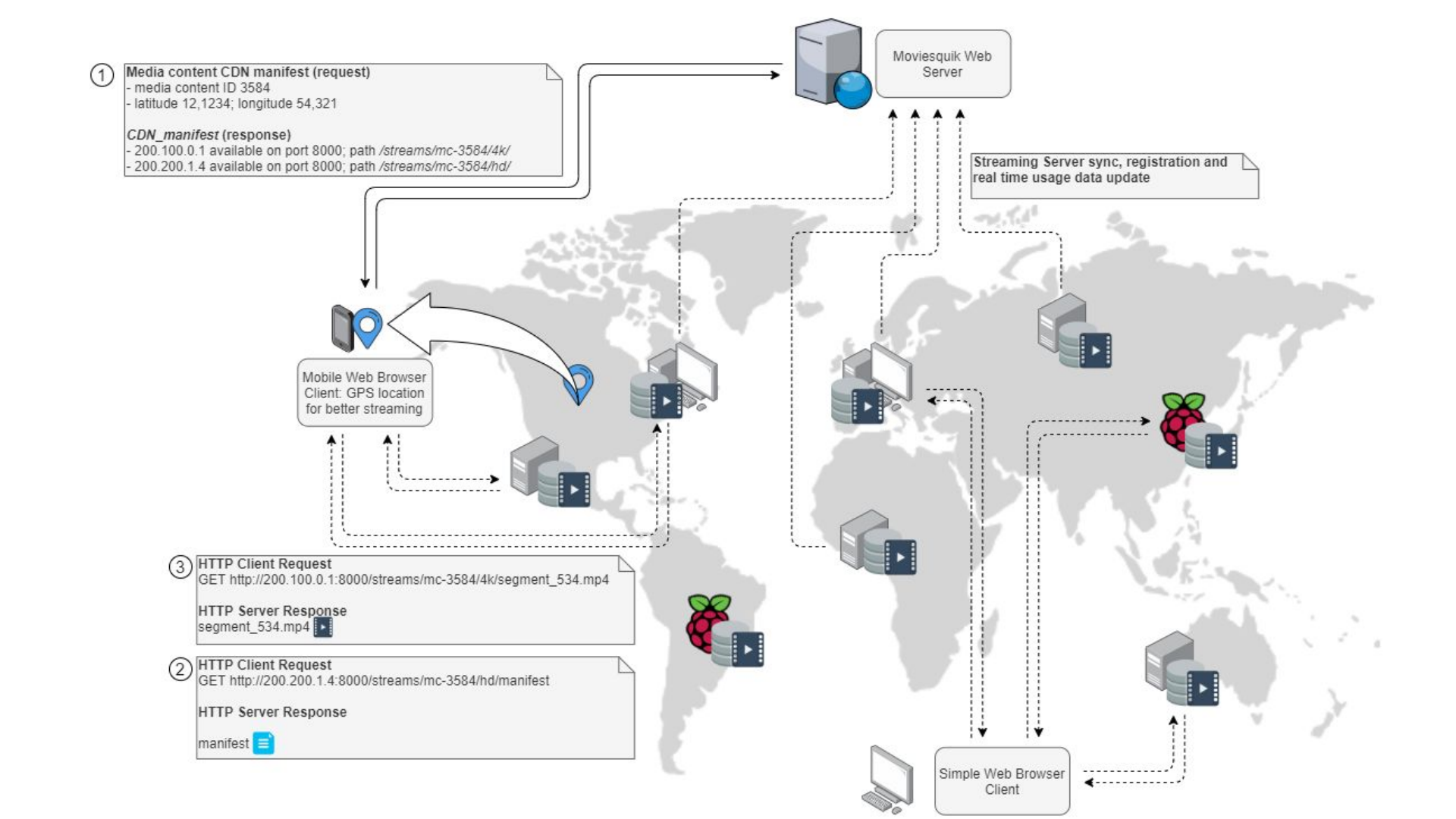
Il Web Server raccoglie queste informazioni sullo stato (periodicamente aggiornato) delle rete peer-to-peer (CDN).

Al momento della richiesta dello streaming di un contenuto da parte di un client, questo consulta il Web Server. Il client, se ne possiede l'accesso, può fornire anche la sua posizione geografica.

Il Web Server consulta lo stato della CDN e crea un *CDN manifest file* contenente l'insieme degli Streaming Servers migliori opportunamente scelti in funzione dell'overhead attuale e del posizionamento geografico rispetto al client (laddove fornito).

Si noti di come sia possibile differenziare lo streaming in funzione della località. Ad esempio, è possibile fornire lo streaming di alcuni contenuti solo in una certa nazione e non in un'altra.

Ottenuto il *CDN manifest file*, il client consulterà direttamente gli Streaming Servers in esso presenti. Tale consultazione avviene utilizzando un **Round-robin Scheduling**.


1
Media content CDN manifest (request)
- media content ID 3584
- latitude 12,1234; longitude 54,321
CDN_manifest (response)
- 200.100.0.1 available on port 8000; path /streams/mc-3584/4k/
- 200.200.1.4 available on port 8000; path /streams/mc-3584/hd/
Moviesquik Web Server
Streaming Server sync, registration and real time usage data update
Mobile Web Browser Client: GPS location for better streaming
3
HTTP Client Request
GET http://200.100.0.1:8000/streams/mc-3584/4k/segment_534.mp4
HTTP Server Response
segment_534.mp4
2
HTTP Client Request
GET http://200.200.1.4:8000/streams/mc-3584/hd/manifest
HTTP Server Response
manifest
Simple Web Browser Client

Streaming Segments

Per effettuare l'upload do un *video file*, esso deve essere opportunamente "installato" in ogni Streaming Server che costituisce la CDN.

L'installazione di un *video file* consiste nel convertire un generico file **.mp4* in una serie segmenti individuali del tipo *segment_0.mp4, segment_1.mp4, …, segment_n.mp4*, ciascuno dei quali ha una durata prestabilita (tipicamente pochi secondi). Le informazioni sulla segmentazione vengono descritte in uno speciale *manifest file* (durata totale, numero di segmenti, mapping segmento-timestamp).

Il *manifest file* verrà richiesto da ogni client all'avvio di ogni streaming. Ottenute le informazioni necessarie sulla segmentazione, il client può procedere al download dei segmenti che necessita (in funzione del timestamp corrente) schedulando le richieste ai servers presenti nel *CDN manifest file* precedentemente ottenuto in seguito alla richiesta di streaming con il Web Server.

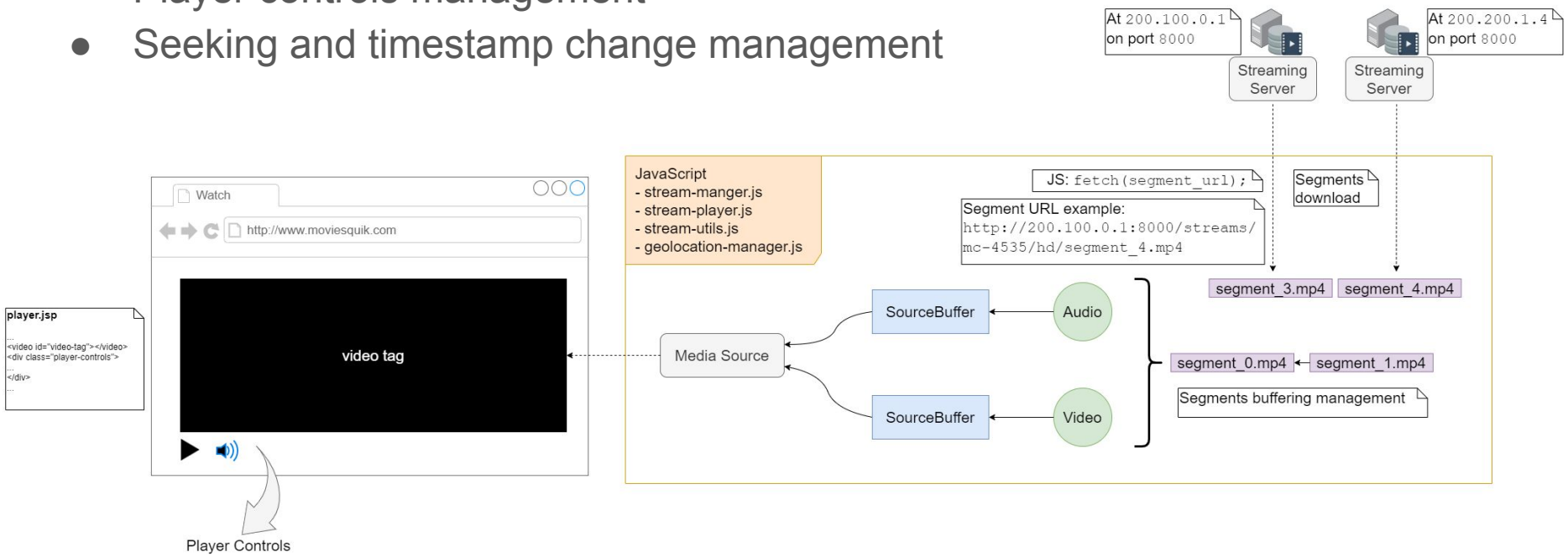
Si noti che il download (effettuato utilizzando il protocollo HTTP) dei segmenti necessari coinvolge tutti gli Streaming Servers con una maggiore distribuzione delle richieste.

L'insieme dei segmenti e il *manifest file* sono collocati nello Streaming Server a partire da una opportuna directory (al path /streams/mc-<MEDIA_ID>). I segmenti possono essere differenziati anche per qualità (4K, HD e LD).

L'installazione di un *video file* su uno Streaming Server può essere effettuata utilizzando un opportuno script di configurazione (incluso nel progetto).

Media Source API (JavaScript)

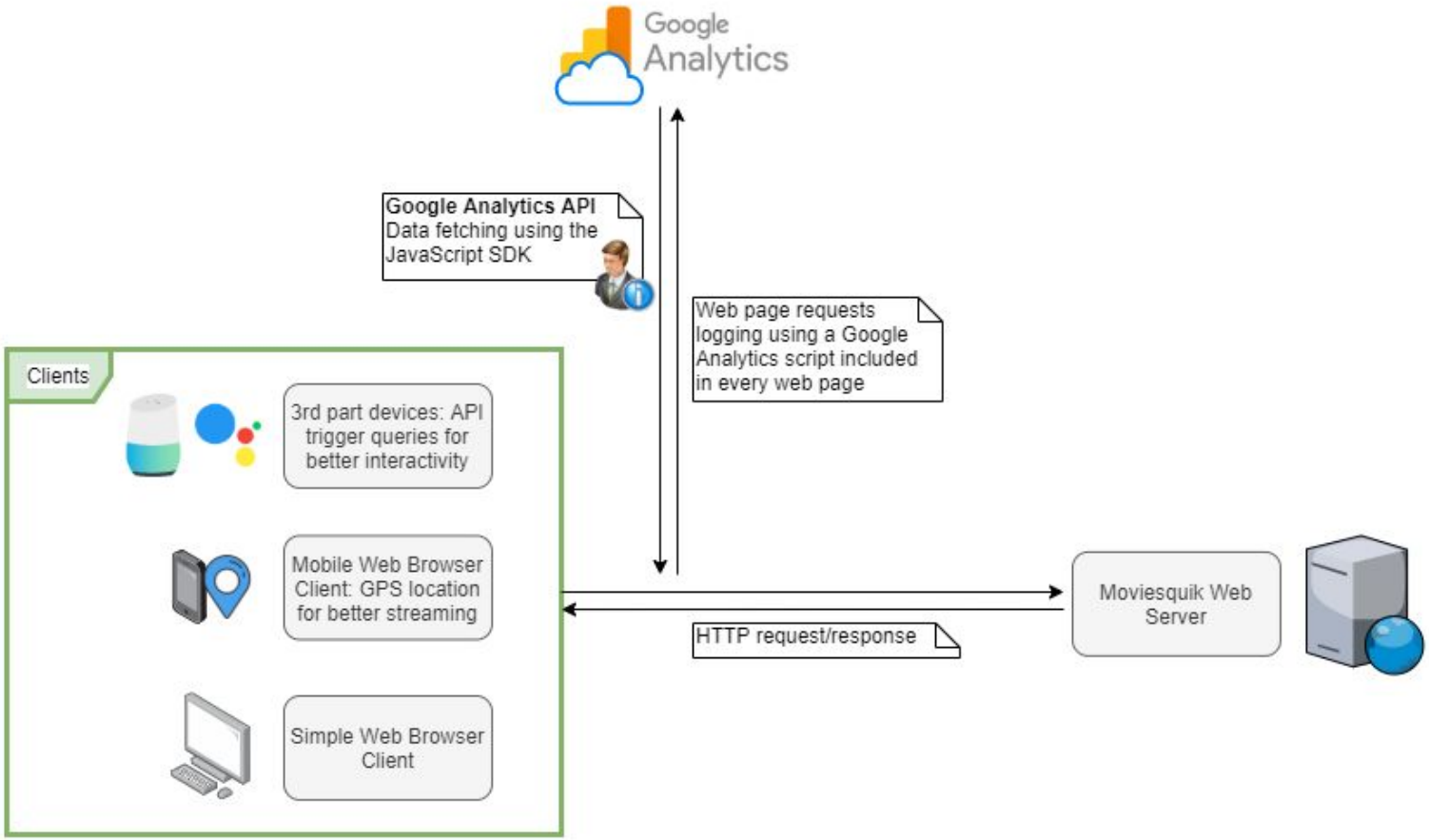
- Segments download using the *fetch(segment_url)* JavaScript function
- Segments buffering management
- Player controls management
- Seeking and timestamp change management



Google Analytics API

- Tracciamento e reporting del traffico
- Ogni Web page include uno script fornito da Google Analytic per il logging delle richieste
- Il report e i dati analitici vengono inclusi nelle pagine Web dedicate all'area Business dell'applicazione utilizzando la apposita SDK JavaScript per l'utilizzo dell'API
- Alcuni dati vengono mostrati attraverso l'utilizzo di grafici quali line charts e map charts. Questi sono inclusi nelle pagine utilizzando specifiche librerie JavaScript come Google Charts e Leaflet-OpenStreetMap
- Alcuni dei dati monitorati e mostrati riguardano: sessions, pageviews, session duration, sessions by country, new visitors, returning visitors, requested pages loggings, used browsers and operating systems

Google Analytics API - 2



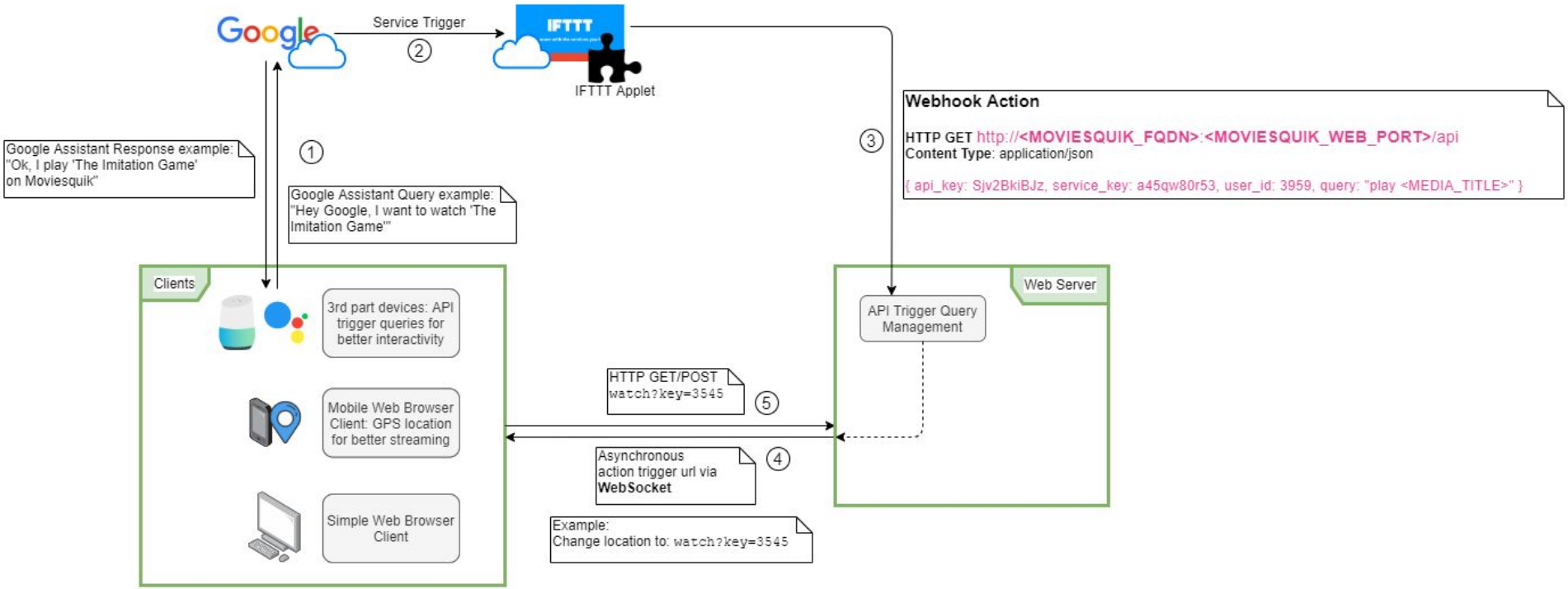
Internal System API for trigger queries

L’applicazione realizzata fornisce la configurazione di una personale API attraverso l’area developer (accessibile solo ai profili non-kid e con abbonamento premium).

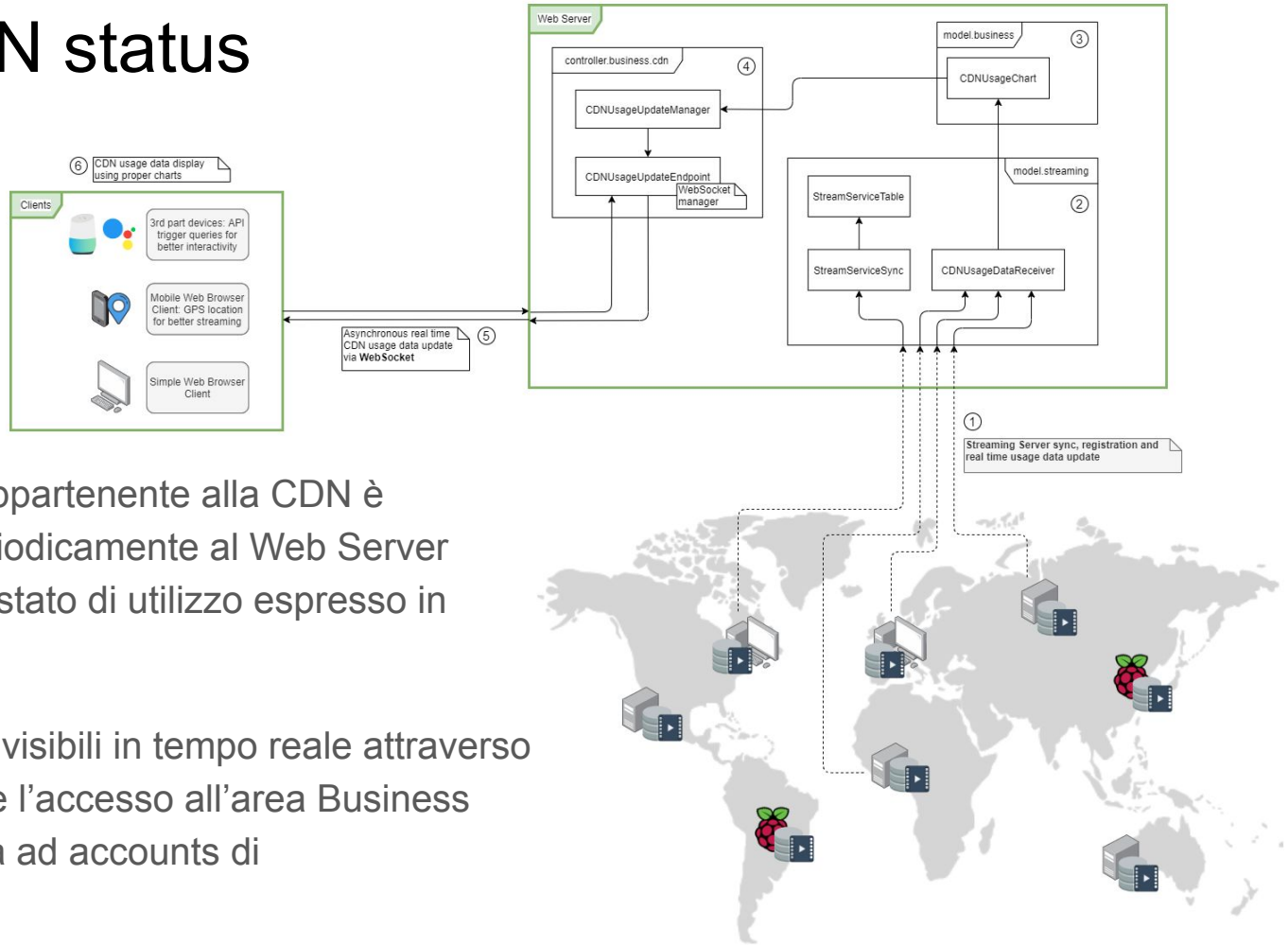
L’API configurata consente quindi l’accesso alla piattaforma mediante servizi di terze parti come l’esecuzione di comandi vocali attraverso l’uso di **Google Assistant** o **Amazon Alexa** (es. “Hey Google, I want to watch ‘The Imitation Game’”).

Per l’implementazione di questa feature è necessario utilizzare il servizio IFTTT per consentire la comunicazione tra Google Assistant e l’applicazione realizzata. Inoltre, è necessaria una comunicazione asincrona tra Web Server e client mediante l’uso dei WebSockets

Internal System API for trigger queries - 2



Real time CDN status



Ogni Streaming Server appartenente alla CDN è progettato per inviare periodicamente al Web Server alcuni parametri come lo stato di utilizzo espresso in requests/second.

Tali informazioni saranno visibili in tempo reale attraverso opportuni grafici mediante l'accesso all'area Business dell'applicazione riservata ad accounts di amministrazione.

WebSocket API

Molte delle funzionalità che prevedono la gestione di eventi asincroni, sono state implementate facendo uso dei WebSockets per consentire una comunicazione interattiva e full-duplex tra browser (client) e Web Server.

- Interactive notifications
- Chats managements (real time messaging, online/offline status, interactive user typing info)
- System API per la gestione di trigger actions (Google Assistant, Amazon Alexa)
- Movie party status update
- Movie party interval management by administrator during multicasting streaming
- Real time CDN (Context Delivery Network) Server usage status charts upload

Media Contents Analytics logging

L’applicazione prevede la raccolta di statistiche sui contenuti e il tracciamento delle preferenze e delle abitudini per ogni singolo utente al fine di fornire una visualizzazione personalizzata di movies e TV shows.

Per la raccolta delle statistiche e il tracciamento delle preferenze vengono monitorate le pagine di ogni contenuto multimediale e del player video tramite opportuni script JavaScript. I dati collezionati riguardano:

- page hits
- page scrolls
- spent time on page
- media views (streaming requests)
- rating e reviews su ogni contenuto
- like dislike media feedbacks

Media Contents Analytics logging - 2

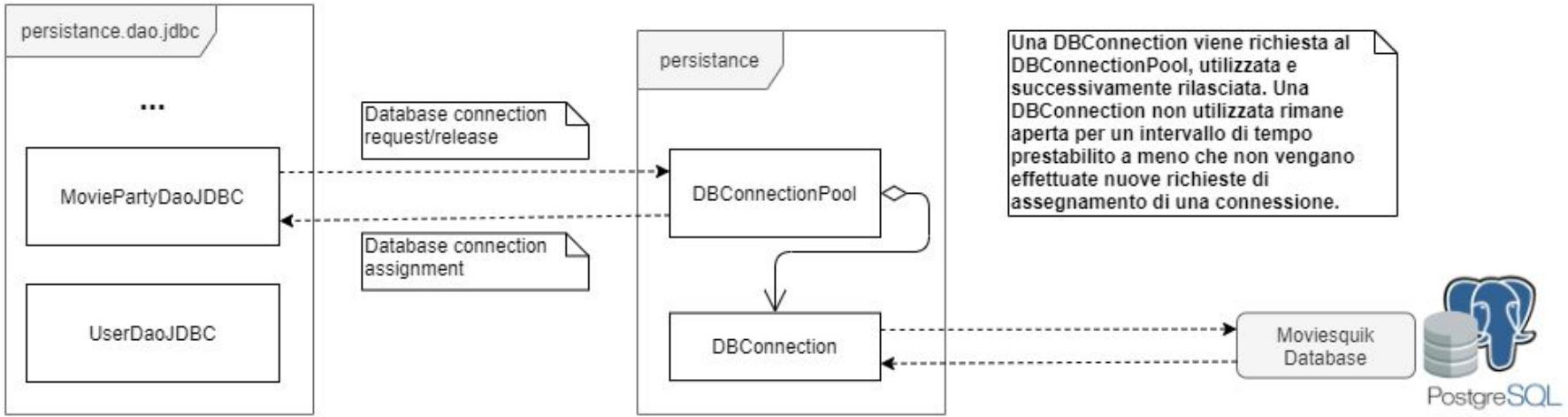
La valutazione di questi dati, tramite opportune query/views/functions SQL, consente il raggruppamento dei contenuti, per ogni utente, secondo le seguenti categorie:

- **suggested for your**: based on computed personal sharing rate* about all genres
- **you may like**: based on inverse computed personal sharing rate* about all genres
- **trending now**: based on short period public sharing rate* (tipically in the past 24 hours)
- **most popular**: based on long period public sharing rate* (tipically in the last week)
- **top rated**: based on user rating and reviews (it also include ratings from the OMDB API)
- **most favorites**: based on user likes and dislikes
- **recently watched**: based on user history

*Il valore di sharing rate viene opportunamente calcolato analizzando i dati monitorati dagli scripts JavaScript su: **page hits**, **page views**, **page scrolls**, **spent time on page** e **media views**

DB Connection Pool

E’ stata data particolare importanza all’allocazione delle connessioni al database mediante l’implementazione di un Connection Pool Manager. La gestione è basata sul riuso di una o più connessioni in caso di frequenti accessi al DB e sulla deallocazione di connessioni rimaste inutilizzate (per un intervallo di tempo prestabilito) diminuendo quindi il numero complessivo di aperture e chiusure di connessione al DB. La richiesta e l’assegnazione di una connessione è Thread-based e migliora considerevolmente le prestazioni di accesso al database.



Web APIs and JavaScript SDKs

- Google Analytics API: analytics website data management
- IFTTT: interaction using third party services
- FacebookAPI: profile registration, login, facebook page posts loading and publishing
- OMDB API: new media contents search and registration
- New York Times Movie Reviews API: media content reviews loading
- Google Charts: analytics data visualization
- Leaflet - Open Street Map: analytics map charts
- Random User Data API: pages filling using random user data

Other main features

- Movie party multicasting and movie interval management by administrator using WebSockets
- Multi-threading background services (MoviePartySyncUpdater, CDNUsageDataReceiver, StreamServiceSync, DBConnectionPool)
- All of the multi-threading background services runs according to thread-safety
- OMDB API for the movie registration and upload in the main database
- User profile image upload
- The application is implemented (server side) using the MVC and DAO patterns
- Most of the page interactions and communications are implemented with asynchronous AJAX calls using, in most cases, JSON as data transfer

CDN Streaming Server scripts

Nel progetto sono inclusi gli scripts di configurazione e funzionamento di uno Streaming Server. Questi includono:

- Uno script Perl (*makestream.pl*) per l'installazione di un video file. Lo script fa uso di alcuni comandi shell quali **ffmpeg**
- Alcuni scripts Python3 (*server.py*, *analytics.py*) per il funzionamento dello streaming server e la raccolta dei dati sulle richieste effettuate. Gli scripts sono in grado di comunicare con il Web Server per la sincronizzazione e l'aggiornamento dei dati di utilizzo.
- Tutti gli scripts sono stati sviluppati ed eseguiti su un sistema Linux, Ubuntu 18.04.4 LTS