Twitter HLD Design

**Twitter** is a [microblogging](https://en.wikipedia.org/wiki/Microblogging) and [social networking service](https://en.wikipedia.org/wiki/Social_networking_service) on which users post and interact with messages known as "tweets".

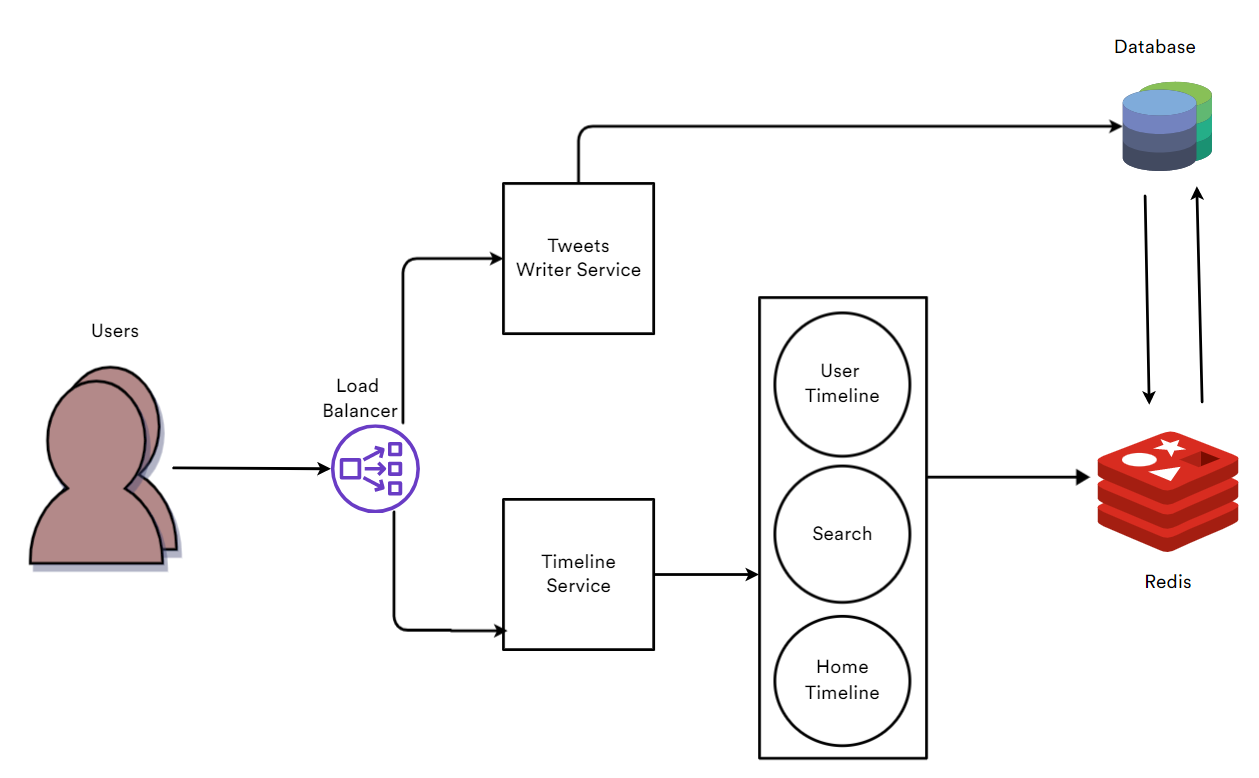
**Features:-**

* The timeline is the main interface for the user to interact with the twitter and it is divided into 3 types

1. Home Timeline : This shows all the tweets to the logged in user of all the personalities or account which the user follows and is the first point of contact for the user.
2. User Timeline : It is responsible for displaying all the tweets and the retweets which the user makes on the application
3. Search Timeline : Filters and displays the tweets according to the input entered by the user into the search bar.

**Vital Obervations and Assumptions :-**

* Twitter is a read heavy application since every second on an average, around 6,000 tweets are tweeted on Twitter & simultaneously 6,00,000 Queries read operations are performed.
* Unlike banking application data consistency is not very important since a delay of 2 to 3 mins is allowed to update data across all the resources.



**Data Flow :-**

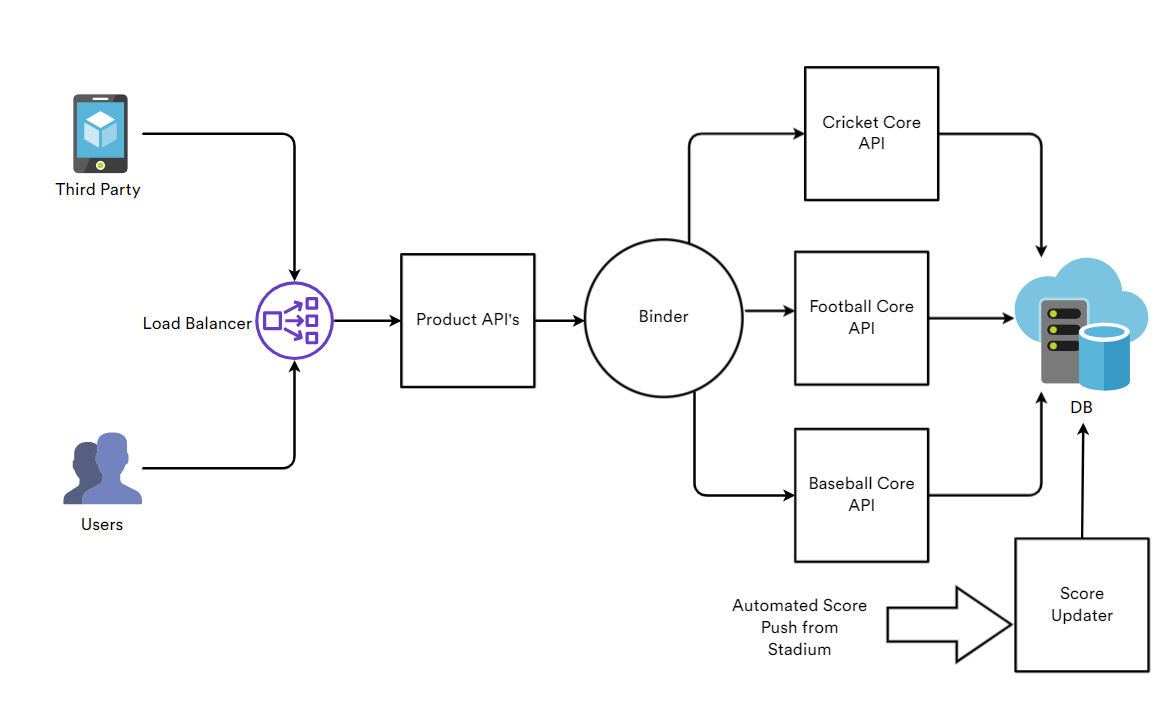
* Load Balancer will distribute the user load across twitter servers around the world.
* The timeline service is responsible for displaying and constructing timelines of the twitter.
* Since we want to have low latency for building the timelines we fetch the data from redis cluster which provides in memory storage of data and helps us to perform caching.
* We also need to maintain all the tweets and data regarding to a particular user like user\_id and tweet\_ids into a main database.
* So, all the data regarding tweets, user info and other data is stored into a main database.
* This main data is then fetched by redis as per the requirement and displayed to user on demand.
* Whenever a user tweets first we store into main DB using Twitter tweets service and then the tweet is distributed across system with help of redis cluster.

Cricbuzz HLD Design

* Cricbuzz is one of the most popular mobile apps for cricket news and scores in [India](https://en.wikipedia.org/wiki/India).
* It features, news, articles and live coverage of cricket matches including videos, text commentary, player stats and team rankings.
* Website traffic details from <https://www.similarweb.com/website/cricbuzz.com/#traffic>



* Crickbuzz has around 421.7 million visitors on an average so we can say that it is read heavy system as the input to these websites is quite less like runs, over, wickets.



* Real time match data is pushed from stadium into score updater service which then stores the data in the database & is distributed over entire app via product and core API’s.
* Data is read by many third party apps and users who use the app directly.
* Two types of API's are usually used

1) Prodcut API

2) Core API

* Since the info received in a single api call will be so much complex and hard to process we divide it into two parts
* Like core API will be for catgories of sports (like football,cricket,baseball,etc)
* Product API's will be for stats, live scores, player info, etc
* Mixer/binder will connect these two api's to serve data faster and handle concurrent requests
* We are able to call multiple core API's with product api's.