

FACEIT Scales to 4m+ Gamers with MongoDB

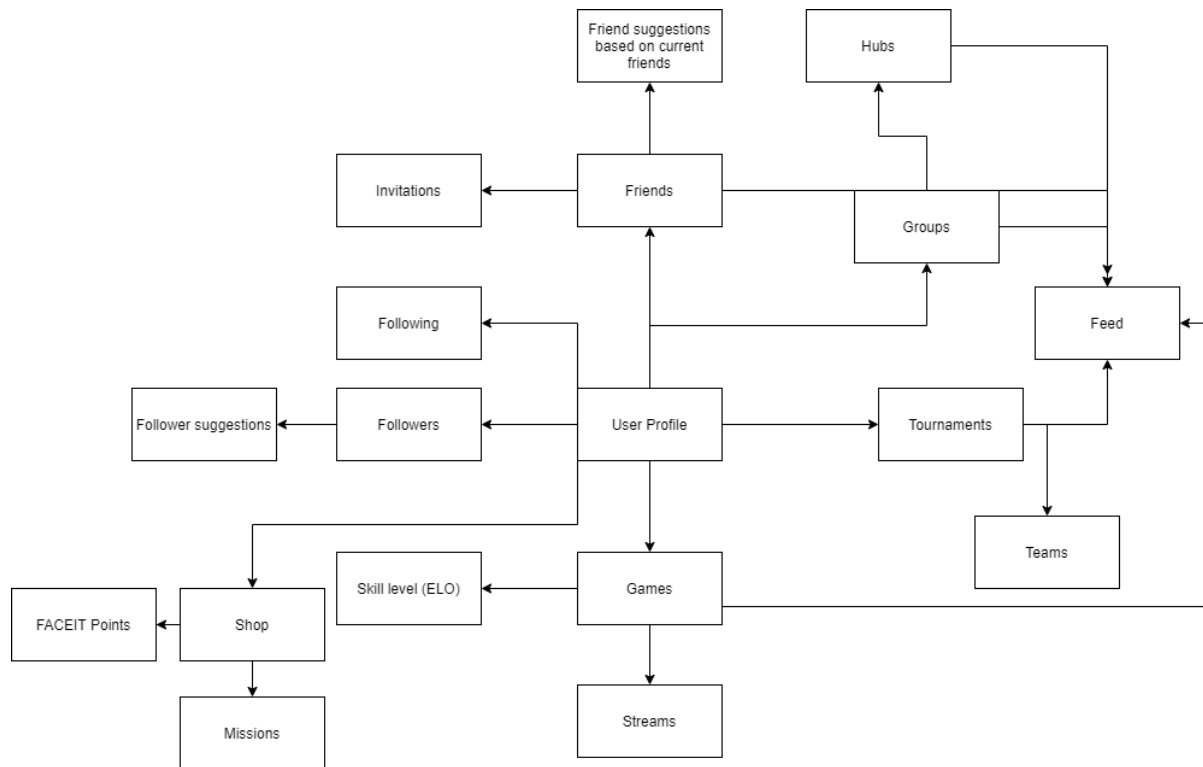
FACEIT is a gaming company that allows users to play certain games with and against other users who have a similar skill level to them. It has become a platform where there are tournaments that are hosted and millions of matches are managed on a monthly basis. FACEIT is used in PvP (player vs player) games such as Counter Strike Global Offensive, DOTA 2, League of Legends etc and elevates the gaming experience for the players that use this service ⁽¹⁾.

One of the main issues that was had with the previous DBMS (Database Management System) was that they would have to constantly update the schema with additional attributes that were related to games, tournaments and features which were implemented. This problem was further enforced through the limitations of relational databases, which could be seen through the constant use of JOIN statements in the DBMS, which uses a large number of resources to perform. As the service grew exponentially, the team realised that the current setup they had regarding the database and storage would be terrible for scalability and would be very difficult to manage efficiently ⁽²⁾.

However, these are not the only features that made MongoDB a viable choice for FACEIT. As the service grew larger and more users joined, it was stated that JOIN operations were very costly and would cause issues for the company going forward. MongoDB was able to combat this in a suitable way through the use of the data models in the document database. Document databases are fundamentally different to traditional relational databases as they actually store and access all the data in one centralised location, rather than different tables as seen in relational databases. This also means that less code has to be written to achieve the same results. Documents are also written in JSON which are far more lightweight than other languages. Due to the nature of JSON, it also means that it is human readable, which again makes it much easier for developers to use.

MongoDB was most definitely the right choice for FACEIT, with how they wanted the data to be stored and manipulated. Using AWS (Amazon Web Services) to store the MongoDB database on the cloud was an excellent choice as AWS is a very powerful cloud service that provides easy scalability. More storage can be purchased as more users join FACEIT, which in turn happened when FACEIT reached 4 million users back in 2016. Not only this but through the optimised compression provided by MongoDB, more than 50% of storage can be saved too.

For FACEIT, it is centred around each individual user, that can play one or more games, participate in tournaments if they please, have multiple friends or no friends, and be a part of groups and communities as well as teams. There are many factors that come into play when trying to organise a user's profile, and with new features being implemented, usually it would be very difficult to add this in to all the pre-existing features and information. But within the document database, all the developers have to do is add to the existing schema, knowing that they do not have to worry about any migration errors and then proceed to automate adding the data to the centralised database.



As you can see there is a lot of information that is attached to a user's profile, even though not all of it will be present on the website when displaying the user's profile. Many of these features, such as the streams, feed and followers are relatively new features and would have been very hard to incorporate into FACEIT's previous database system, but was easily added into the document database provided by MongoDB.

A common software practice that is currently incorporated by FACEIT is running two replica sets that store the same data, using the same version of MongoDB and the software provided by AWS. This means that in case one set fails and data is lost, it can be restored using the other.

In conclusion, FACEIT was able to migrate from a traditional DBMS to a cloud-based service provided by MongoDB in order to improve the scalability of their service by solving the problem of the limiting factor, being the storage of user's data and the impact it had on performance.

Presentation: <https://www.youtube.com/watch?v=1JvPWKEZ2Pk>

References:

(1) - <https://www.youtube.com/watch?v=ncWN-EbbhhU>

(2) - <https://www.mongodb.com/blog/post/faceit-scales-to-4m-gamers-with-mongodb-and-cloud-manager#>

(3) - <https://www.mongodb.com/document-databases>