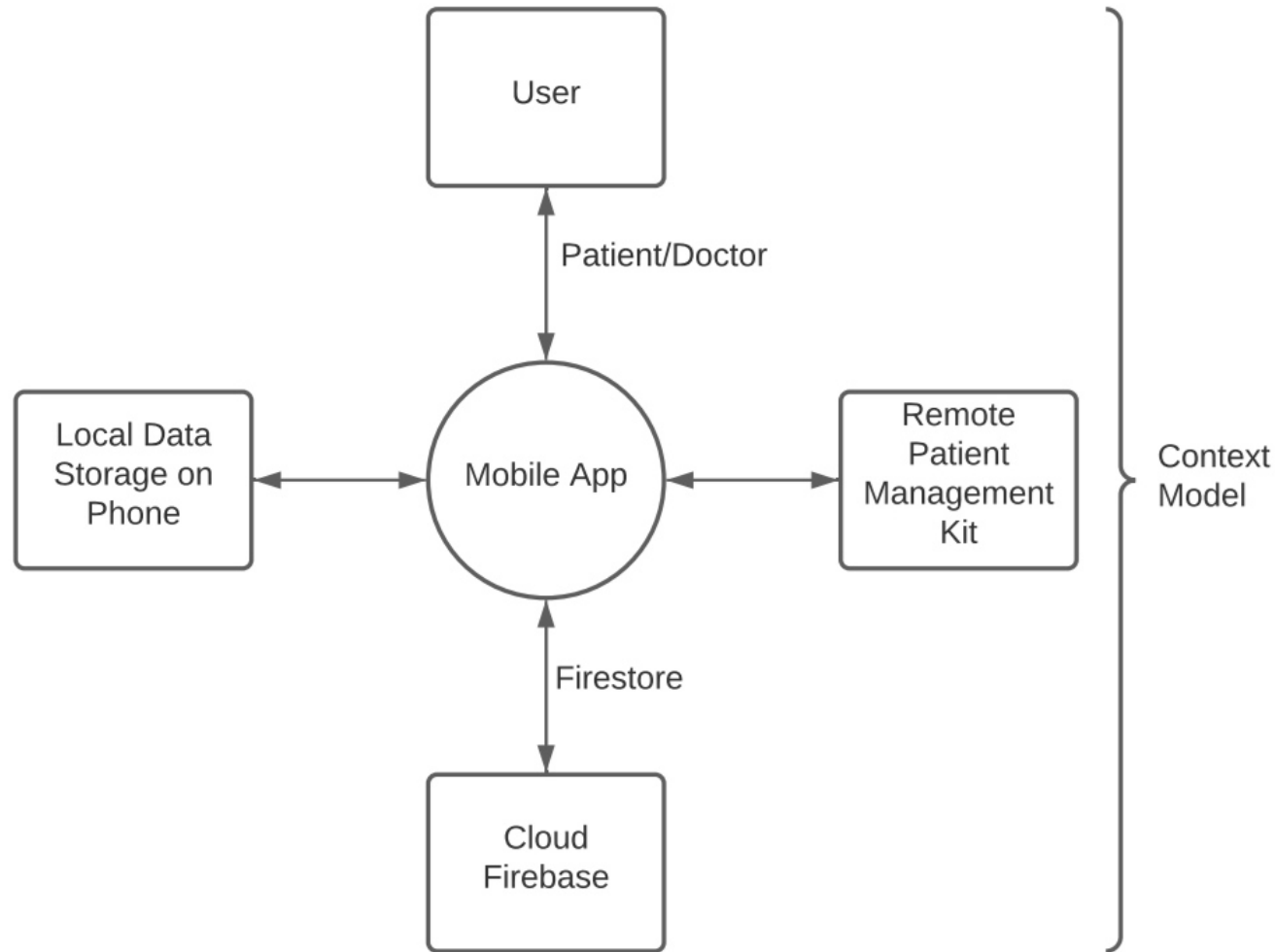




Firestore and Firebase

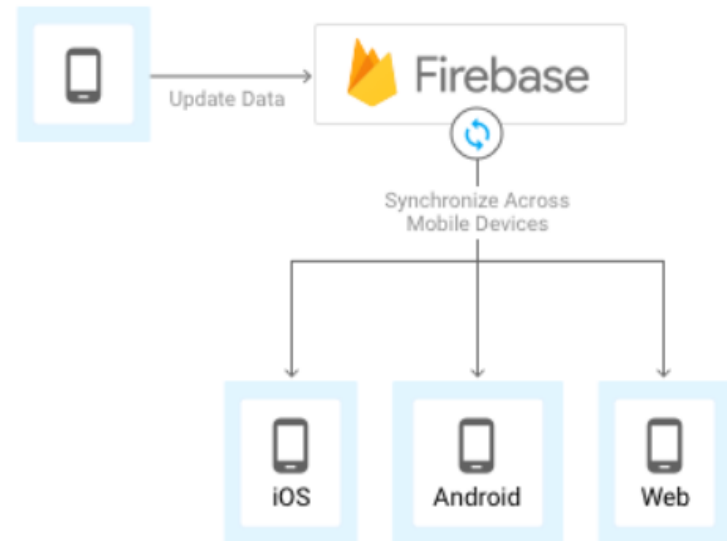
By team Technocrats

Context of our application



What is Firebase?

Firebase is a fully managed platform for building iOS, Android, and web apps that provides automatic data synchronization, authentication services, messaging, file storage, analytics, and more.

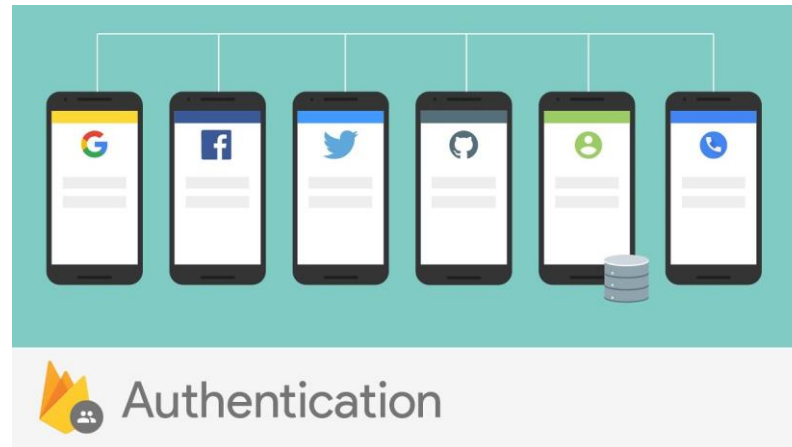


Features

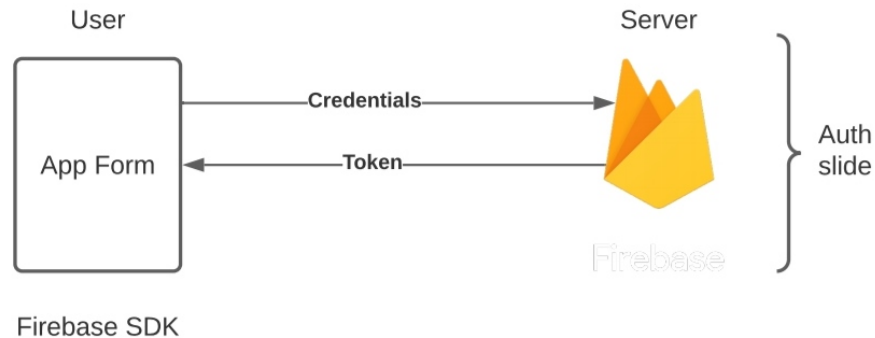
- Realtime
- Offline
- Accessible from client devices
- Scalable across multiple databases

User Authentication

- Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.
- Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to the app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.



Working of user authentication



- To sign a user into the app, first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, pass these credentials to the Firebase Authentication SDK. The backend services will then verify those credentials and return a response to the client.
- After a successful sign in, one can access the user's basic profile information, and can control the user's access to data stored in other Firebase products. They can also use the provided authentication token to verify the identity of users in their own backend services.

Cloud Firestore

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. Like Firebase Realtime Database, it keeps the data in sync across client apps through realtime listeners and offers offline support for mobile and web which enables building of responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud Platform products, including Cloud Functions.




Features of Firestore

- Flexibility
- Expressive querying
- Realtime updates
- Offline support
- Designed to scale

Working

- Cloud Firestore is a cloud-hosted, NoSQL database that iOS, Android, and web apps can access directly via native SDKs.
- Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries. Documents support many different data types, from simple strings and numbers, to complex, nested objects. You can also create sub collections within documents and build hierarchical data structures that scale as the database grows


 users

 alovelace

first : "Ada"

last : "Lovelace"

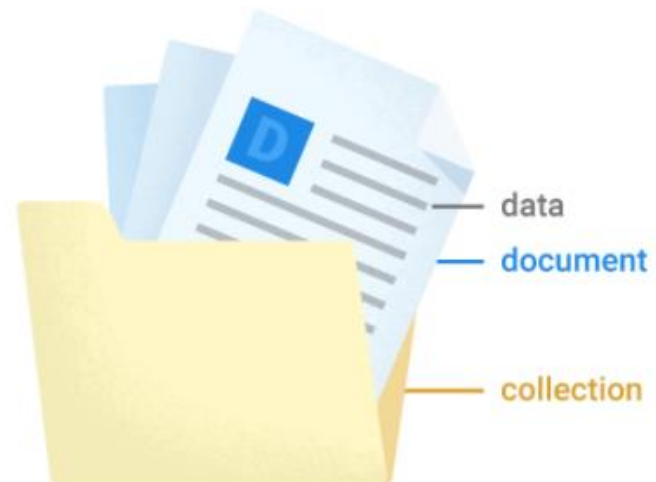
born : 1815

 aturing

first : "Alan"

last : "Turing"

born : 1912

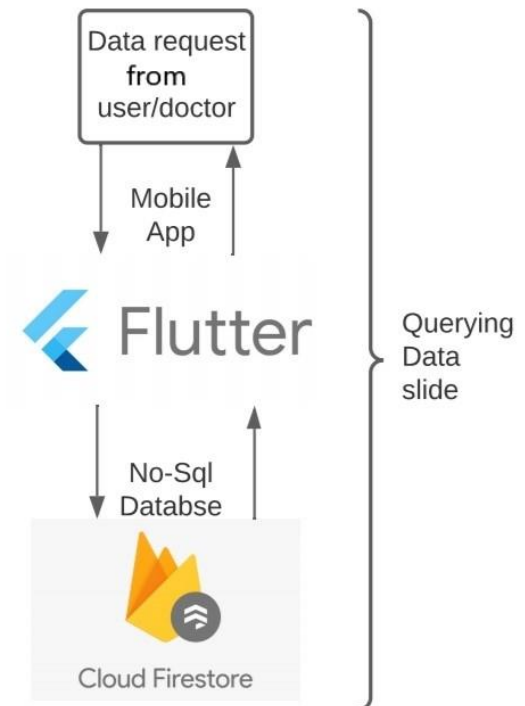


Data Querying

There are two ways to retrieve data stored in Cloud Firestore. Either of these methods can be used with documents, collections of documents, or the results of queries:

- Call a method to get the data.
- Set a listener to receive data-change events.

When a listener is set, Cloud Firestore sends the listener an initial snapshot of the data, and then another snapshot each time the document changes.



Security Rules

Cloud Firestore offers robust access management and authentication through two different methods, depending on the client libraries used.

- For mobile and web client libraries, use Firebase Authentication and Cloud Firestore Security Rules to handle serverless authentication, authorization, and data validation.
- For server client libraries, use Identity and Access Management (IAM) to manage access to your database.

Firebase Security Rules stand between the data and malicious users. One can write simple or complex rules that protect app's data to the level of granularity that the specific app requires.

Working

Firestore Security Rules work by matching a pattern against database paths, and then applying custom conditions to allow access to data at those paths. All Rules across Firebase products have a path-matching component and a conditional statement allowing read or write access. One must define Rules for each Firebase product is used in the app.

For Cloud Firestore, Rules use the following syntax:

```
service <<name>> {  
  // Match the resource path.  
  match <<path>> {  
    // Allow the request if the following conditions are true.  
    allow <<methods>> : if <<condition>>  
  }  
}
```

Pros and Cons

- Application can be shared across multiple clients
- Queries with limited sorting and filtering functionality can be performed. Cloud firestore assures automatic scaling and can handle 1 million concurrent connections and 10,000 writes/second.
- Helps in the easy storing and retrieval of dynamic content.
- It offers integration to Google Ads, AdMob, the Play Store, Data Studio. Analytics, crashing reports are also provided by the firebase so that the development teams can stay focused on enhancing the user experience.
- Firebase for android app development completely relies on a flat hierarchy of nested data. Relational queries cannot be dealt with ease using firebase.
- Difficult to perform complex querying.
- Data migration in the application becomes difficult with firebase since it is not like the general SQL database or the object-relational mapping.
- Security rules are limited in firebase that makes it very difficult in building enterprise platforms over them.