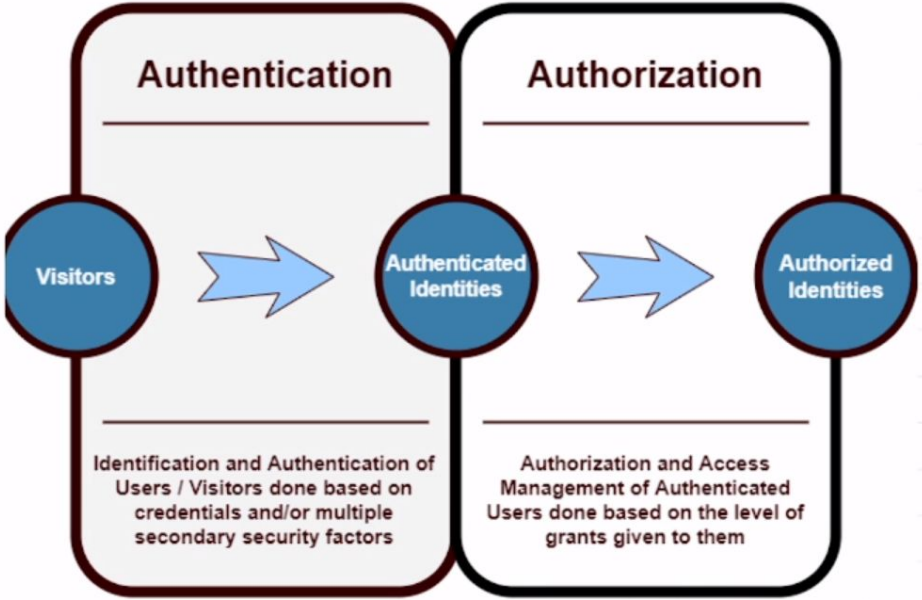
Simple React Firebase Authentication

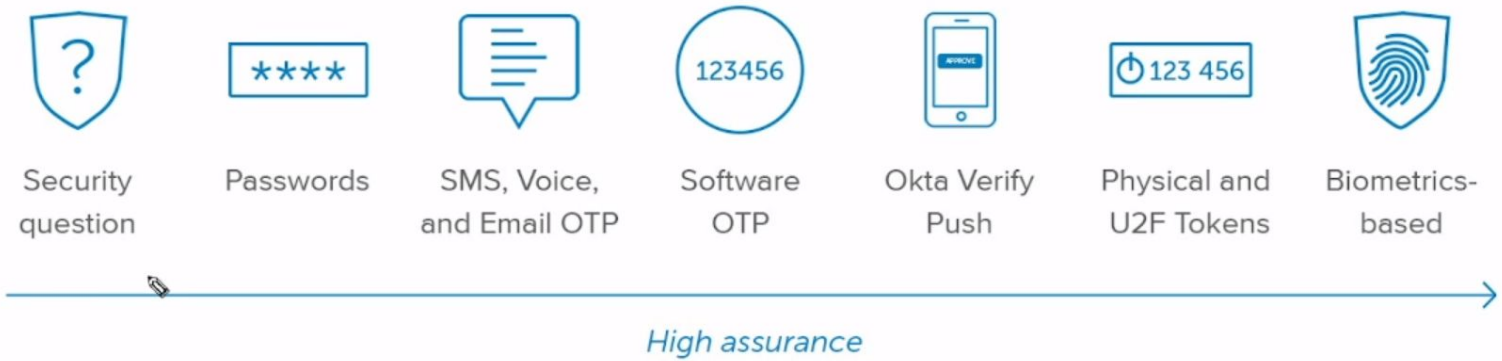
Authentication vs Authorization



Authentication vs Authorization

Authentication	Authorization
• Determines whether users are who they claim to be	• Determines what users can and cannot access
• Challenges the user to validate credentials (for example, through passwords, answers to security questions, or facial recognition)	• Verifies whether access is allowed through policies and rules
• Usually done before authorization	• Usually done after successful authentication
• It usually needs the user's login details	• While it needs user's privilege or security levels
• Generally, transmits info through an ID Token	• Generally, transmits info through an Access Token

Authentication Methods



Interview questions

- What is Firebase?
- **Is firebase frontend or backend?**
- What is firebase architecture?
- **What are the features of firebase? / What are the tasks you can accomplish with firebase?**
- Differences between firebase and mongodb.
- **Have you ever used firebase database (real time database)?**
- Can you briefly explain the github authentication process with firebase?
- **Which method is used to Sign-in the user in Firebase Email/Password authentication?**
- Authentication VS Authorization
- **Can you tell me names of 3 authentication method?** (hint: Sms/ email code, voice, password, fingerprint, face verification)
- Which authentication methods have you ever used for your project purpose?

What is Firebase?

Is firebase frontend or backend?

What is Firebase?

Is firebase frontend or backend?

What is Firebase?

Firebase is a comprehensive and powerful mobile and web application development platform provided by Google. It offers a wide range of tools and services that enable developers to build feature-rich and scalable applications quickly and easily. Firebase provides a backend-as-a-service (BaaS) infrastructure that includes services such as real-time database, authentication, hosting, storage, cloud messaging, remote config, and more.

Is firebase frontend or backend?

Firebase is a backend-as-a-service (BaaS) platform provided by Google. It offers a range of cloud-based services that primarily focus on providing backend functionality for mobile and web applications. This includes services such as real-time database, authentication, cloud messaging, storage, hosting, and more.

What is firebase architecture?

What are the features of firebase? / What are the tasks you can accomplish with firebase?

What is firebase architecture?

Firebase follows a serverless architecture, where much of the backend infrastructure and operations are managed by Google, allowing developers to focus on building frontend (client-side) applications without having to worry about managing their own servers. The architecture of Firebase typically includes the following components:

Differences between firebase and mongodb

Have you ever used firebase database (real time database)?

Differences between firebase and mongodb

1. **Database Type:** Firebase is a NoSQL, cloud-based, real-time database,
2. Backend-as-a-Service **(BaaS)** vs Database-as-a-Service (DBaaS): Firebase is a BaaS platform that provides a complete backend solution, including authentication, real-time database, cloud messaging, hosting, and more,
3. **Real-time Data Sync:** Firebase provides real-time data synchronization, allowing multiple clients to stay in sync with the latest data changes in real-time, making it well-suited for applications that require real-time collaboration or live data updates.
4. **Scalability and Flexibility:** Firebase is a fully managed, serverless solution provided by Google, which means it handles scaling, load balancing, and other operational aspects automatically

1. MongoDB is a NoSQL, on-premises or cloud-based, document-oriented database.
2. MongoDB is a standalone database that requires developers to build their own backend infrastructure.
3. MongoDB, on the other hand, does not provide built-in real-time data synchronization.
4. on the other hand, requires manual configuration and management for scaling and load balancing, making it more suitable for applications that require fine-grained control over the database.

Why we are user firebase database (real time database)?

Real-time updates: Firebase Realtime Database provides real-time updates, allowing changes made to the data to be instantly reflected across all connected clients or devices without the need for manual refreshes or polling. This makes it ideal for applications that require real-time data synchronization and updates, such as chat applications, real-time monitoring applications, and collaborative tools.

Scalability: Firebase Realtime Database is designed to handle large amounts of data and concurrent users, making it suitable for applications with high levels of concurrent reads and writes. It can scale automatically to accommodate increased load without the need for manual configuration or setup.

Offline data access: Firebase Realtime Database supports offline data access, allowing applications to continue functioning even when offline, and automatically synchronizing data changes once connectivity is restored. This makes it suitable for applications that need to work seamlessly both online and offline, such as mobile applications.

Can you briefly explain the github authentication process with firebase?

- Which method is used to Sign-in the user in Firebase Email/Password authentication?

Can you briefly explain the github authentication process with firebase?

Setting up a Firebase project:

Enabling GitHub as a Sign-in provider:

Implementing GitHub authentication in your application:

Authenticating with GitHub:

Handling the authentication callback:

Accessing user information:

Securing access to resources:

Authentication VS Authorization

Can you tell me names of 3 authentication method? (hint: Sms/ email code, voice, password, fingerprint, face verification)

Which authentication methods have you ever used for your project purpose?

Authentication VS Authorization

1. Authentication and authorization are often used interchangeably,
2. Authentication is the process of verifying the identity of a user or entity, typically through the use of credentials such as username and password, biometric information, or other means. It ensures that the user or entity is who they claim to be
3. Authorization, on the other hand, is the process of granting or denying permissions to authenticated users or entities to access specific resources or perform certain actions within a system or application.
4. authentication is the process of verifying the identity of a user or entity, while authorization is the process of granting or denying permissions to authenticated users or entities based on their privileges or roles.

1. they are two distinct concepts in the realm of security and access control in software applications.
2. the authentication process is successful, the user or entity is granted access to the system or application.
3. Authorization ensures that authenticated users or entities have appropriate permissions to perform actions or access resources based on their privileges.
4. authorization determines what you are allowed to do. Both authentication and authorization are important components of secure access control in software applications.

Thank you!