Fullstack Development

Authentication / Authorization

Part 3: Persisting auth's state

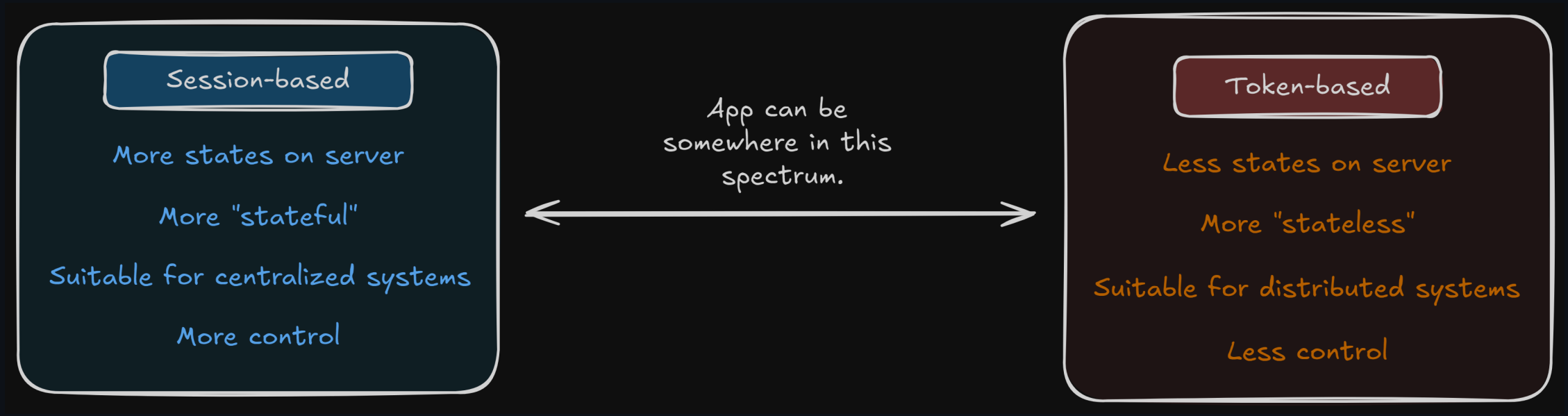
Session based

- Server is responsible for creating and maintaining the user's authentication state (i.e. in a database).
- After user sign-in, the server sets a cookie that contains the session ID and sends it to the browser.
  - The browser will include it in all further requests.
  - The server will use the cookie to identify the current user session from the database.

Token-based

- A "token" is a cryptographically signed piece of data that contains information about the authenticated user and their access permissions.
- The server will only have to verify the validity of the token rather than having it stored in a database.
  - Reduces the amount of state that needs to be stored on the server.
- While other token formats exist, JSON Web Tokens (JWTs) have become the prevailing standard for token-based approach.

Not one or another



Most systems use both approches in various degrees.

Please do not do this.

- It is temping to go 100% stateless using token-based approach (JWT) to avoid dealing to storing information on server.
- Be aware of these dangers.

> If you don't have database table storing `auth` in your system, there is something wrong.

Considering token-based approch

- You are losing control over user's state.
  - You are making your system less secured.
- If you are concerned about hitting database too much,
  - have you considered `redis`?
- Modern solutions for tokens are as, if not more, complex as the session-based one.
  - Acess/refresh tokens
  - Token rotation
  - Allowed/revoked list