Authorization with JWT Claims

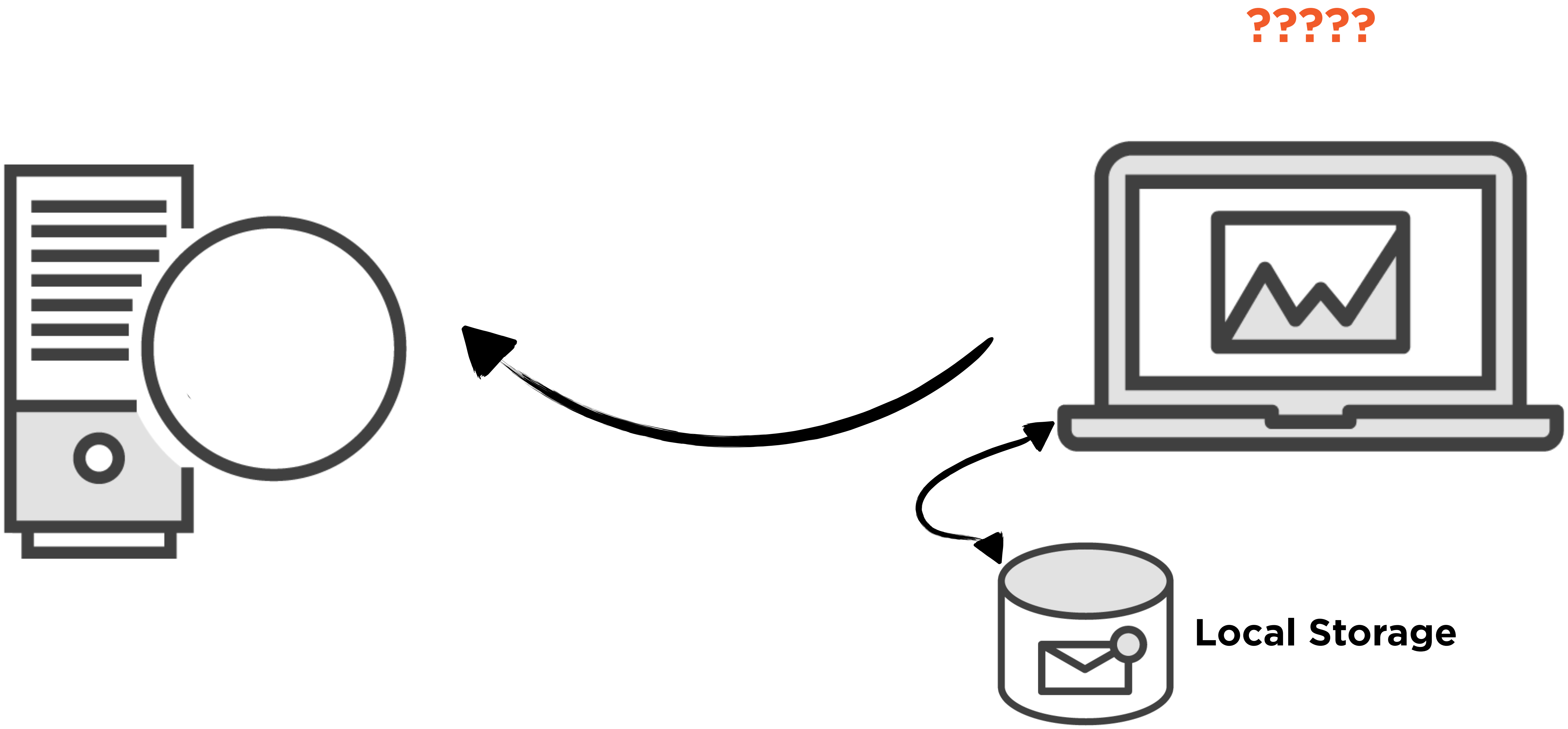
Deeksha Sharma
CO-FOUNDER AND ENGINEER
@deekshasharma25   www.bonsaiilabs.com

Agenda

- Token handling in Client Code
- Update UI with token claims
- APIs more robust and secure

Store, Send and Use JWT - Client

JWT in Auth Header



Response Data

username/role

Response Data

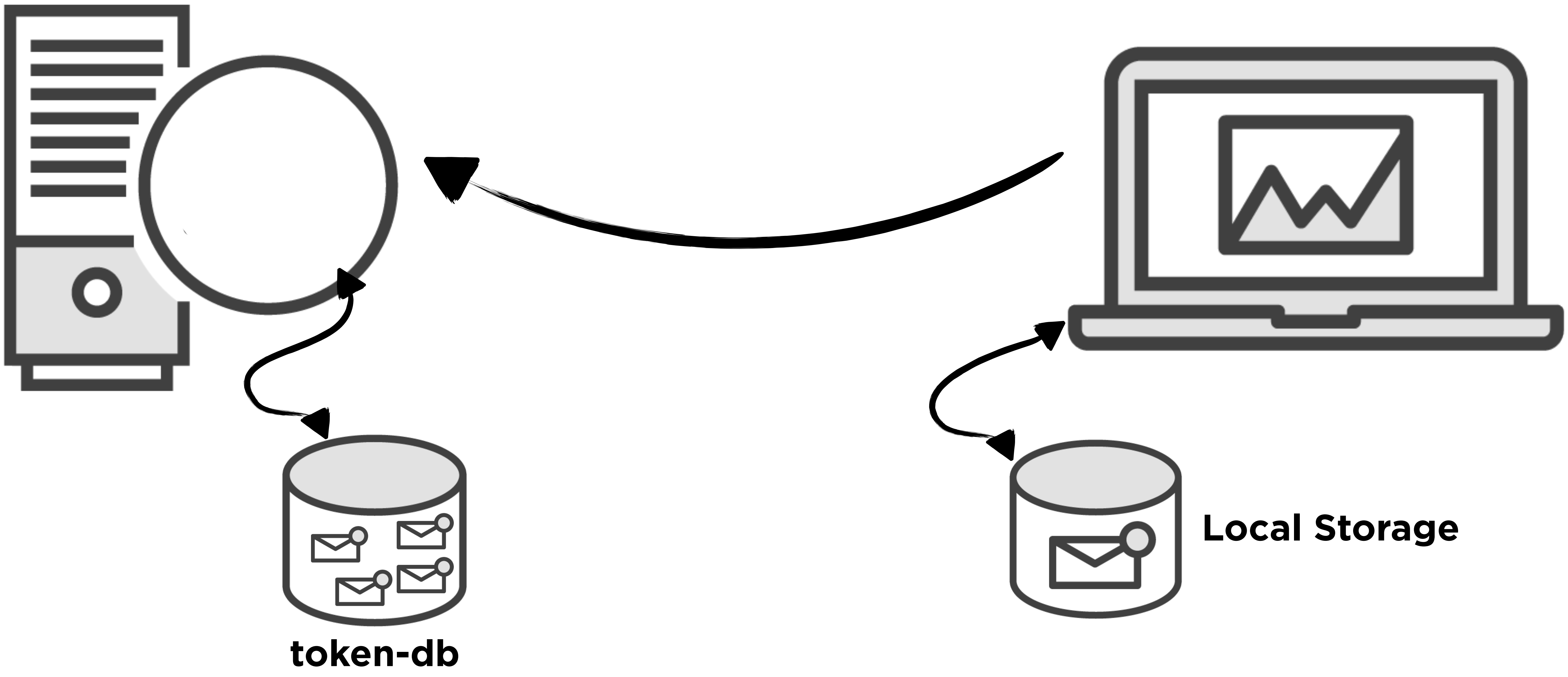
audience claim

Decode Token

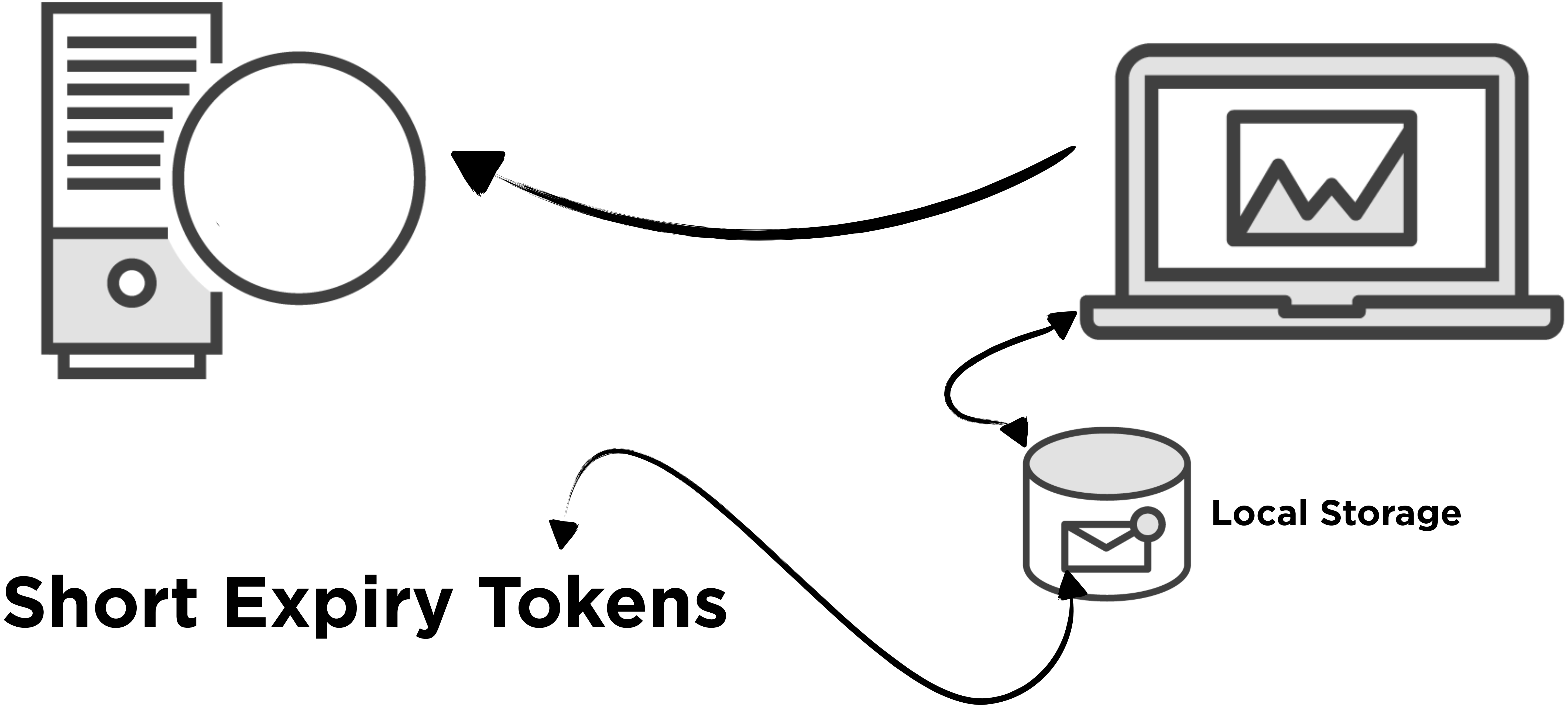
audience claim

Make APIs Robust and Secure

Check Issued Tokens



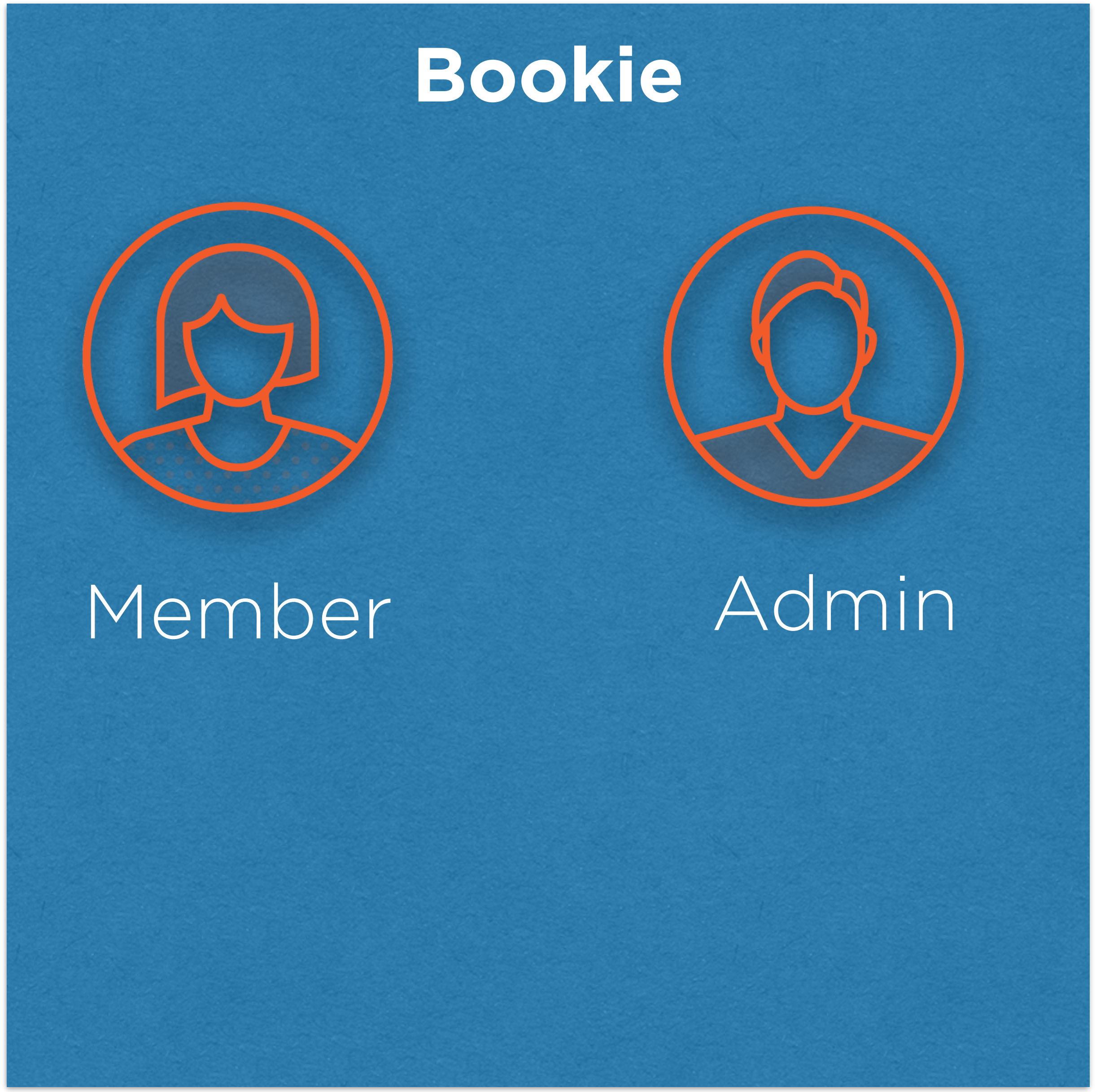
Check Issued Tokens



Local Storage

Short Expiry Tokens

Not Production Grade

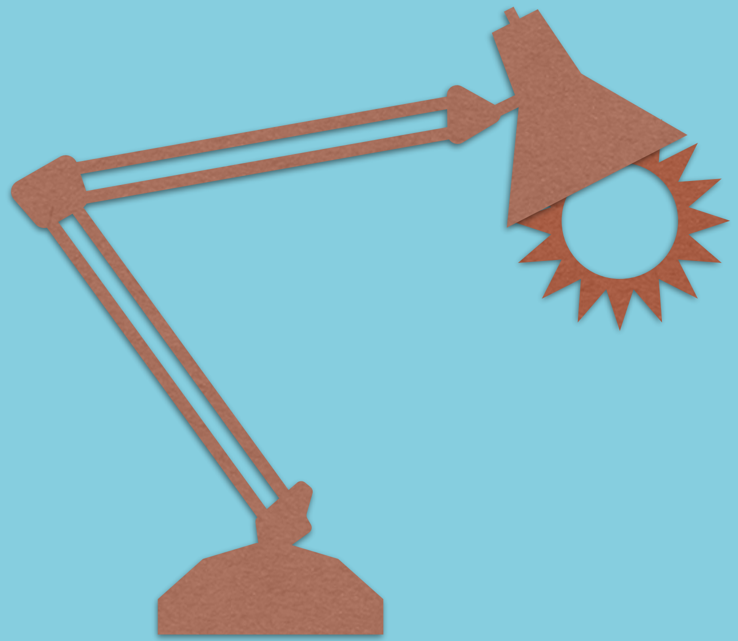


Bookie

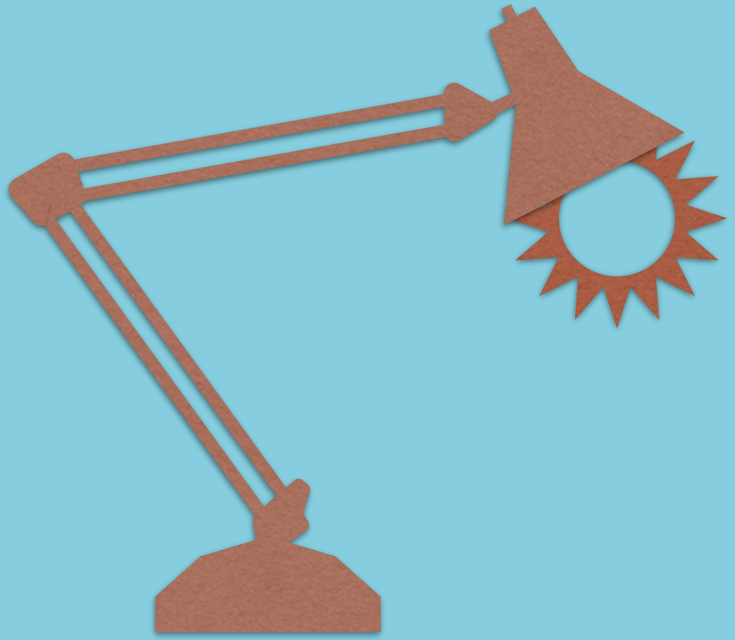
Member

Admin

- /login 🔒
- /logout 🔒
- /books 🔒
- /favorite 🔒
- /users 🔒
- /book 🔒

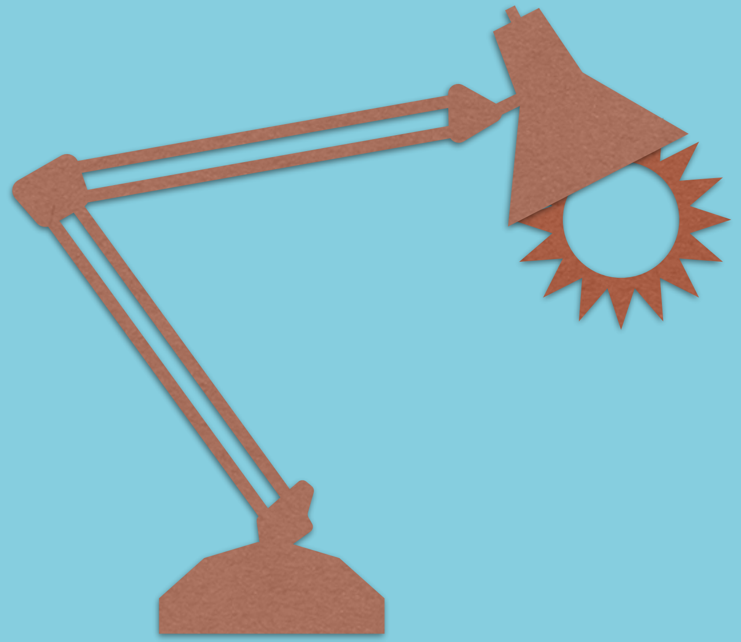


Always Verify Signature presence in the JWT



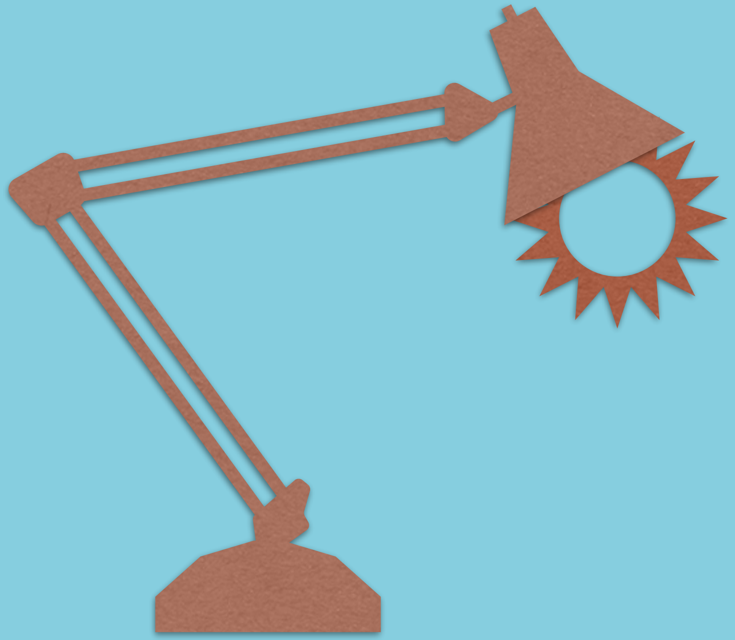
Reject Expired Tokens

Reduce expiry in **variables.env** *and check*

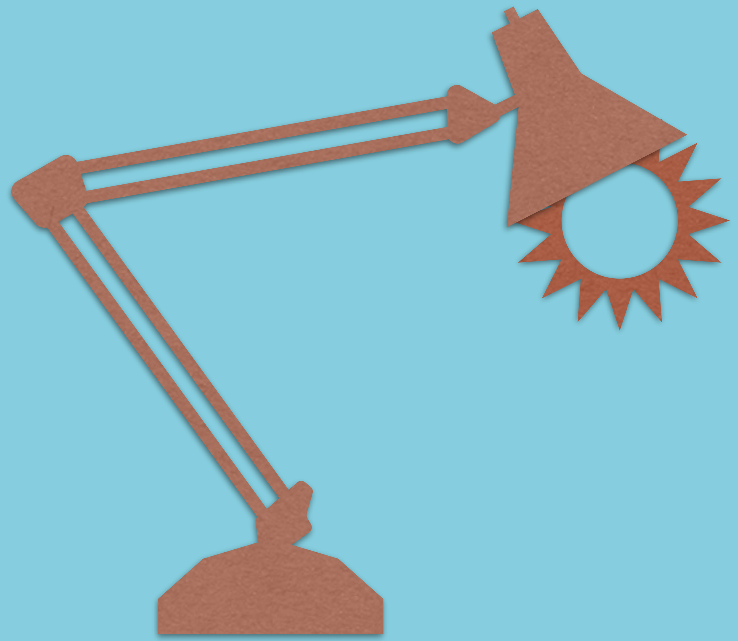


Test Code with Tampered Tokens

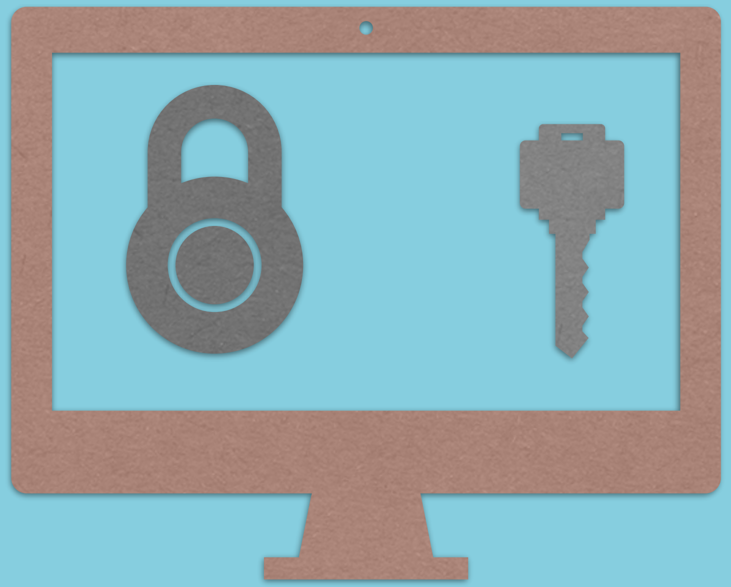
Tamper token using Chrome Dev Tools or POSTMAN

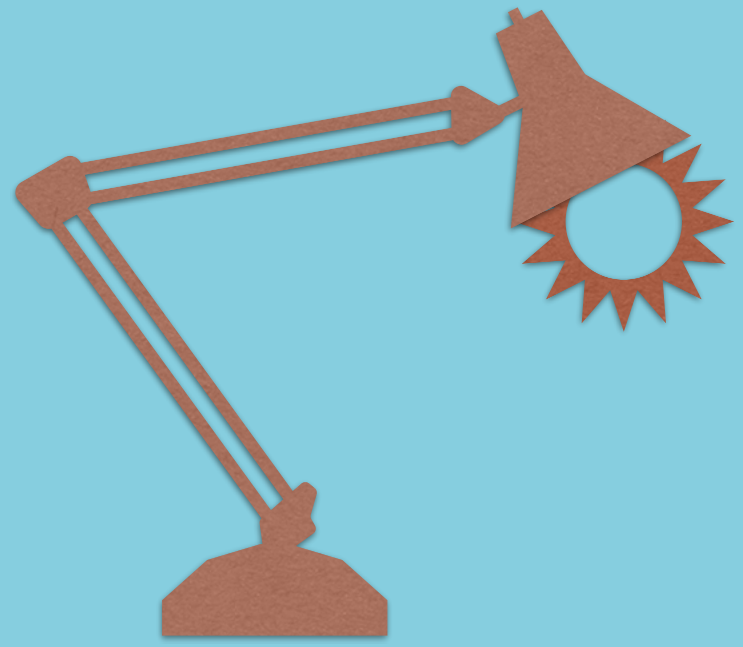


Make Front End Routes Inaccessible

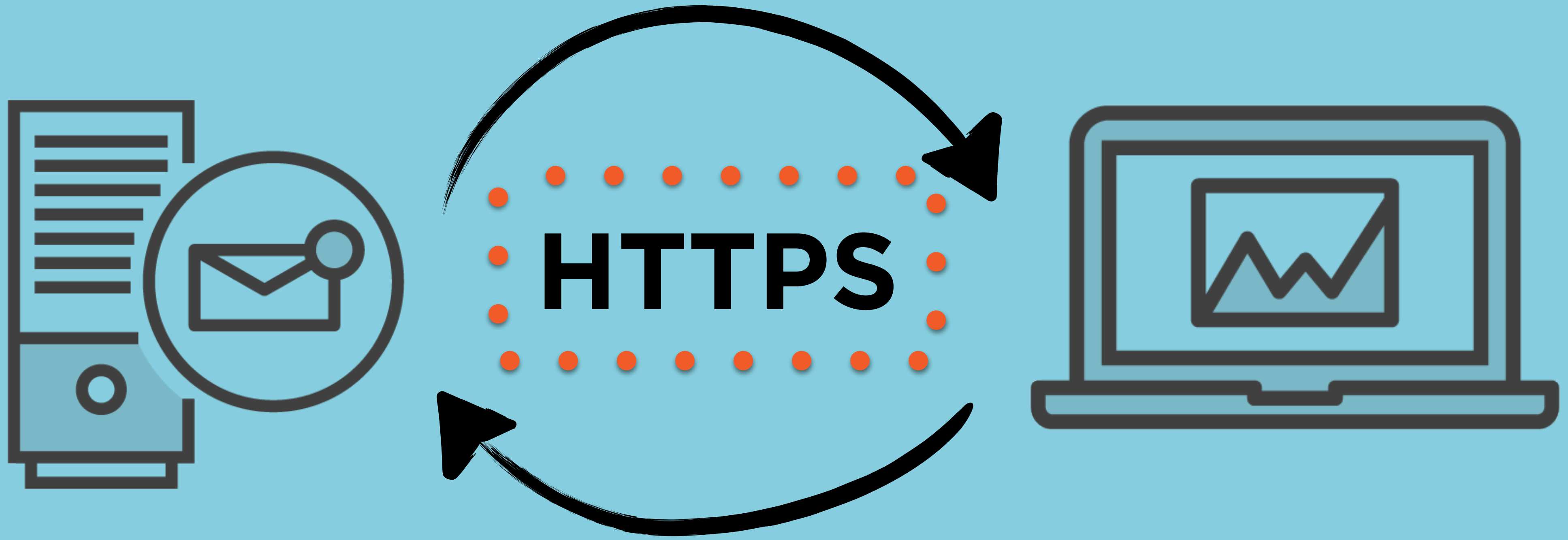


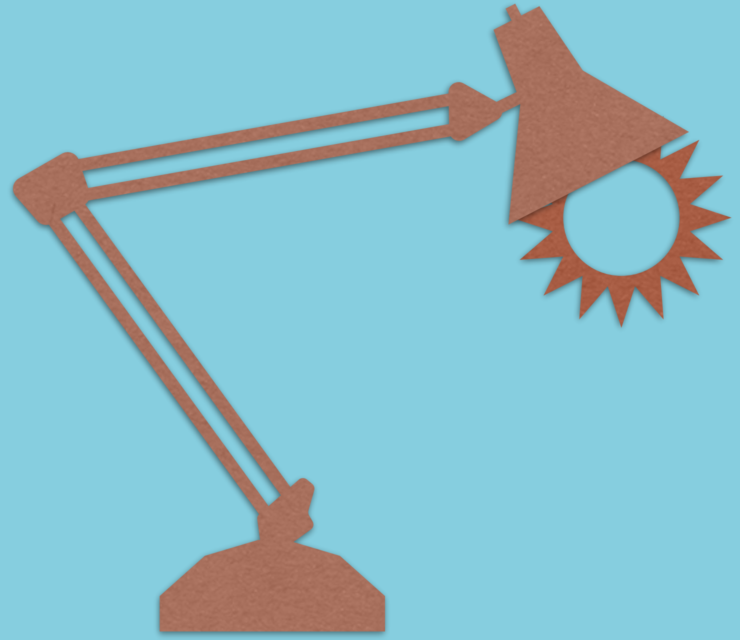
JSON Web Encryption (JWE)





Only HTTPS Please!!



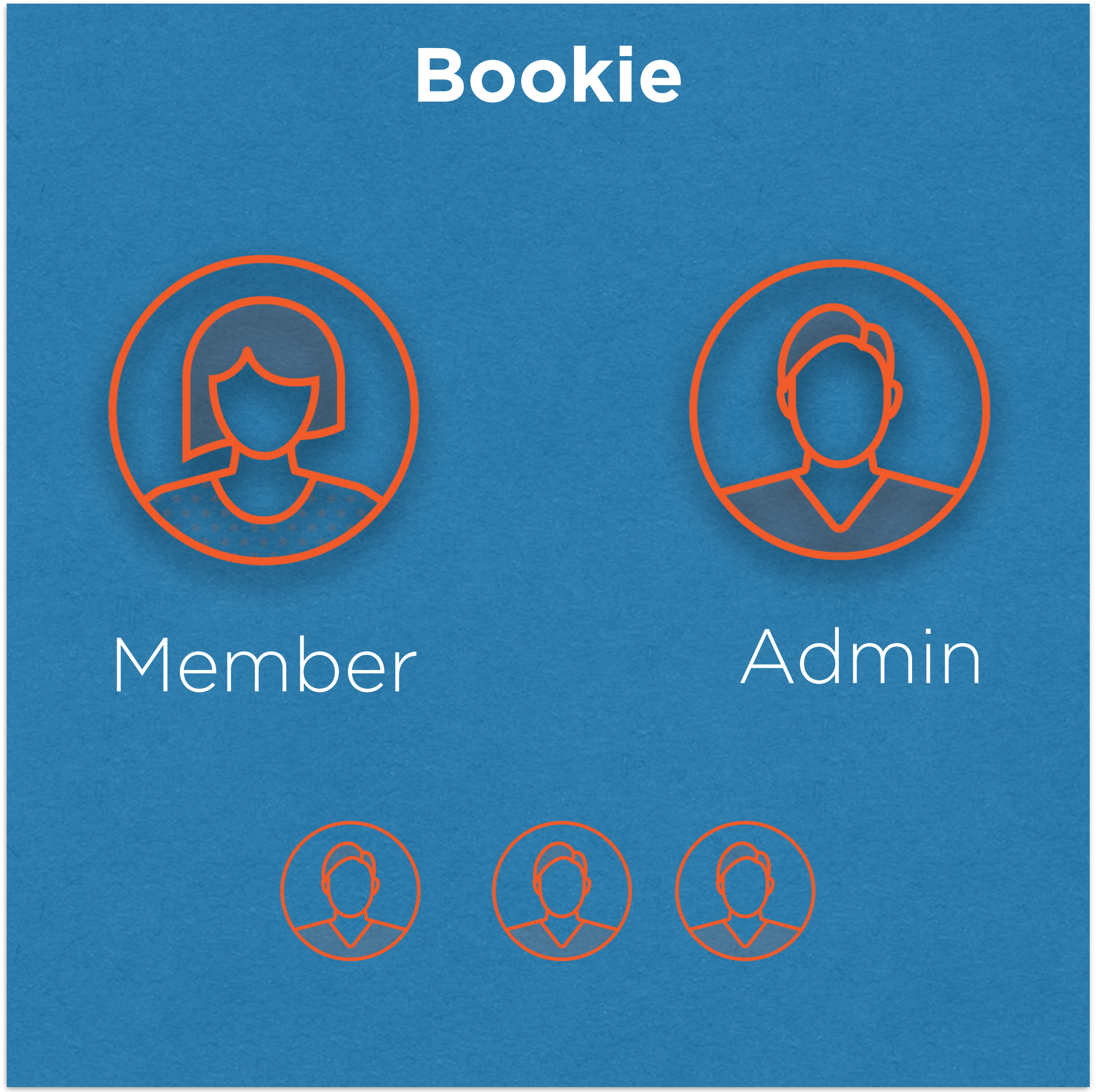


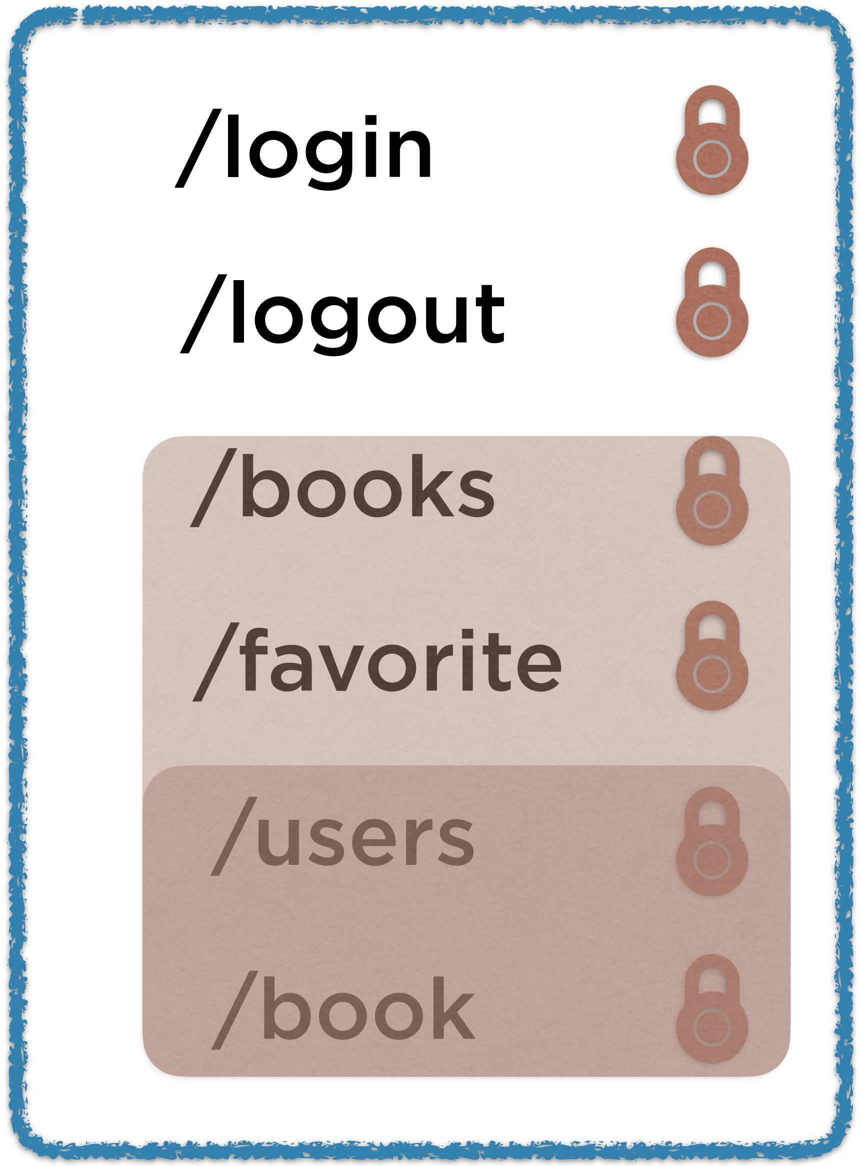
Validate User Inputs in Client Code

Disallow Empty Form Fields

Check for Duplicate Books

Content Type





/login

/logout

/books

/favorite

/users

/book

Summary

- Token handling in Client Code
- Updated UI elements with token claims
- Make APIs robust and secure