Your Next Week

Tuesday April 21	Wednesday April 22	Thursday April 23	Friday April 24
6:30 PM **— DUE Class 11 Lab** **— DUE Class 12 Reading** — Class 12A — Instructor Syncs	*6:30 PM* — Class 12B — Instructor Syncs *MIDNIGHT* **— DUE Class 12 Learning Journal**	*6:30 PM* — Co-Working — Instructor Syncs	
Saturday April 25	**Sunday April 26**	**Monday April 27**	**Tuesday April 28**
6:30 PM **— DUE Class 12 Code Challenge** **— DUE Class 12 Lab** **— DUE Class 13 Reading** — Class 13 *MIDNIGHT* **— DUE Class 13 Learning Journal**	*MIDNIGHT* **— DUE Class 12-13 Feedback**	*6:30 PM* — Career Coaching Workshop #1 Continued (Mandatory)	*6:30 PM* **— DUE Class 13 Lab** **— DUE Class 13 Code Challenge** **— DUE Class 14 Reading** — Class 14A

Lab 11 Review

Class 12

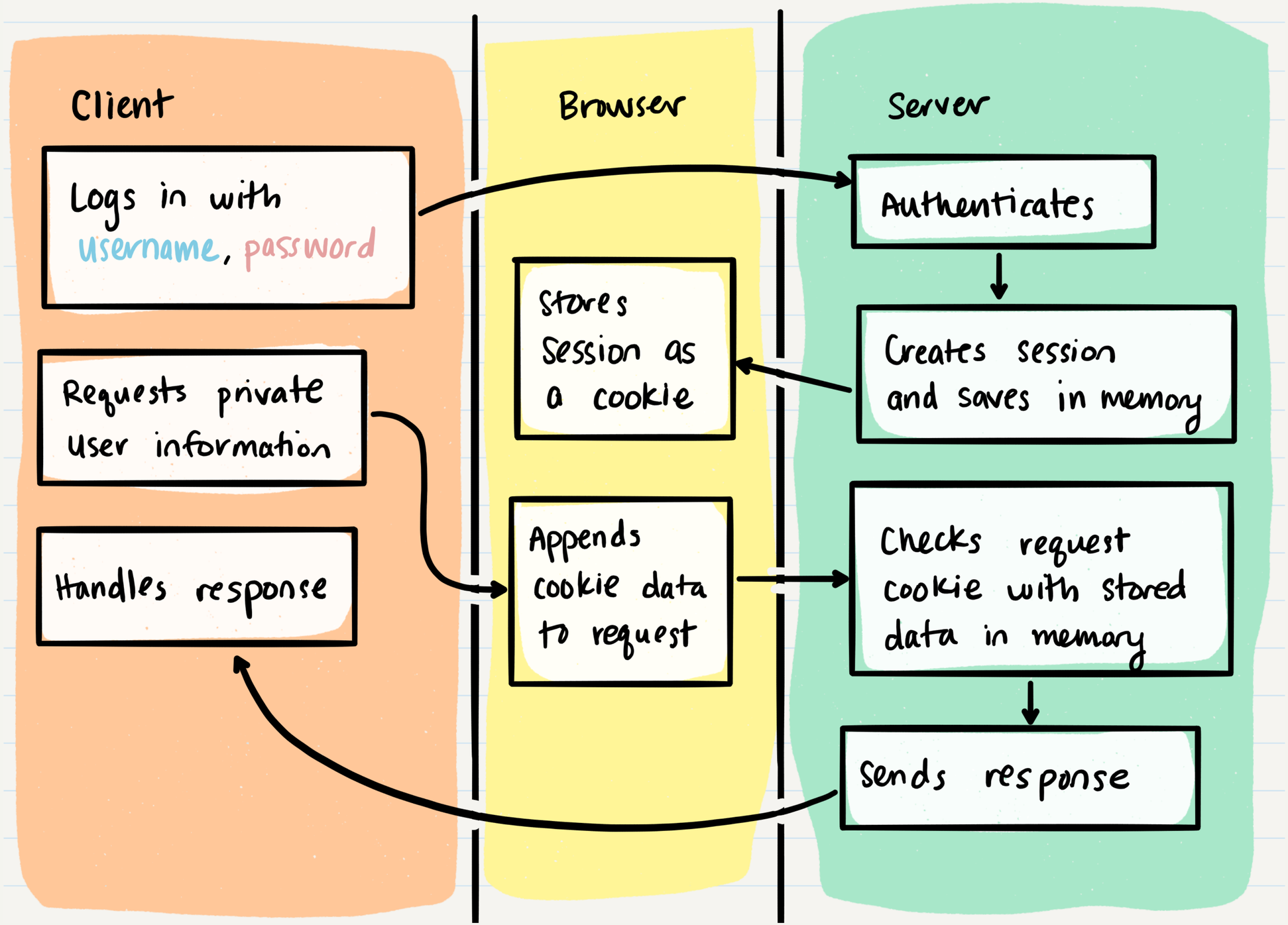
OAuth

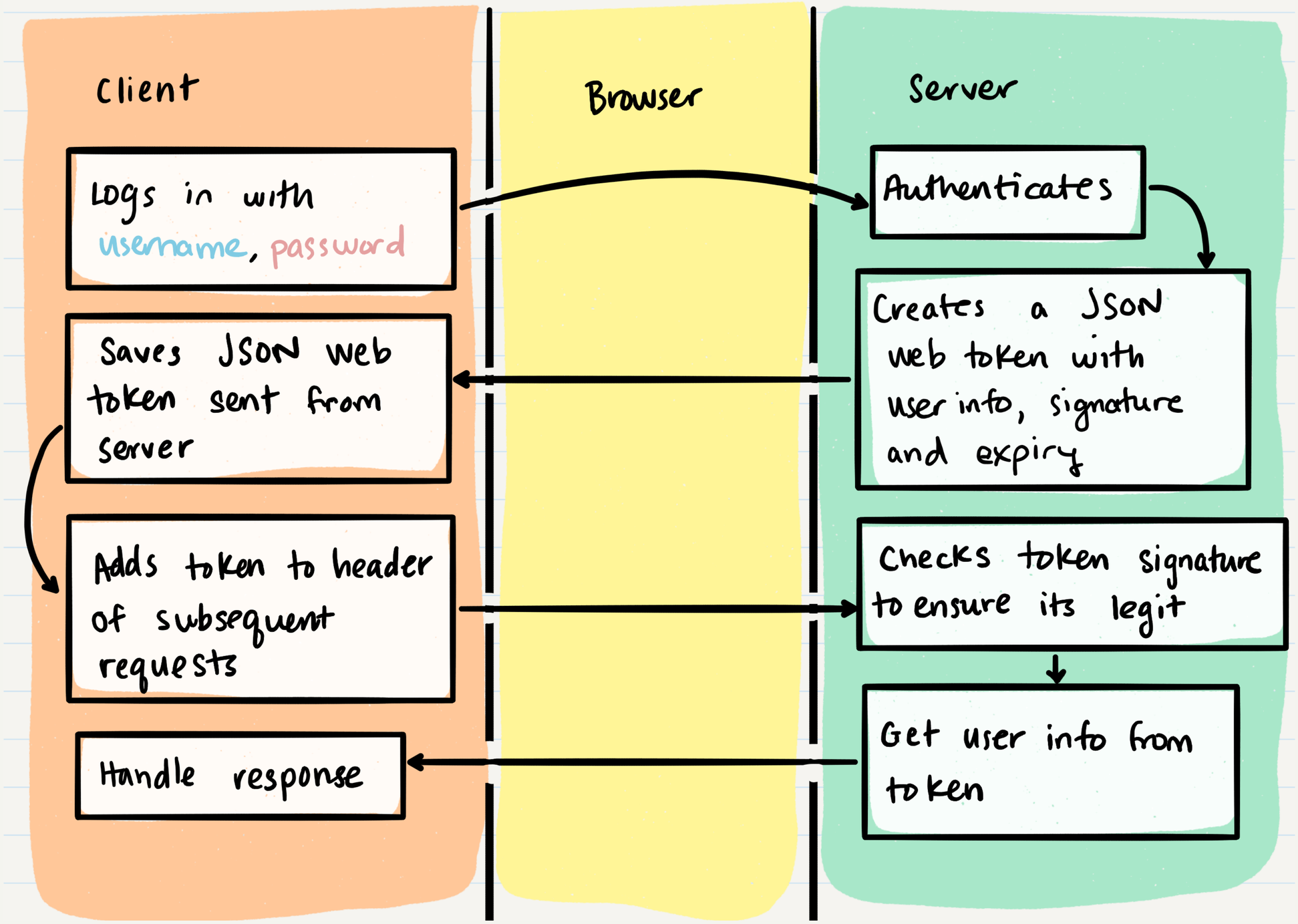
seattle-javascript-401n16

But first…
Things we didn’t cover from Class 10!

Sessions and Tokens

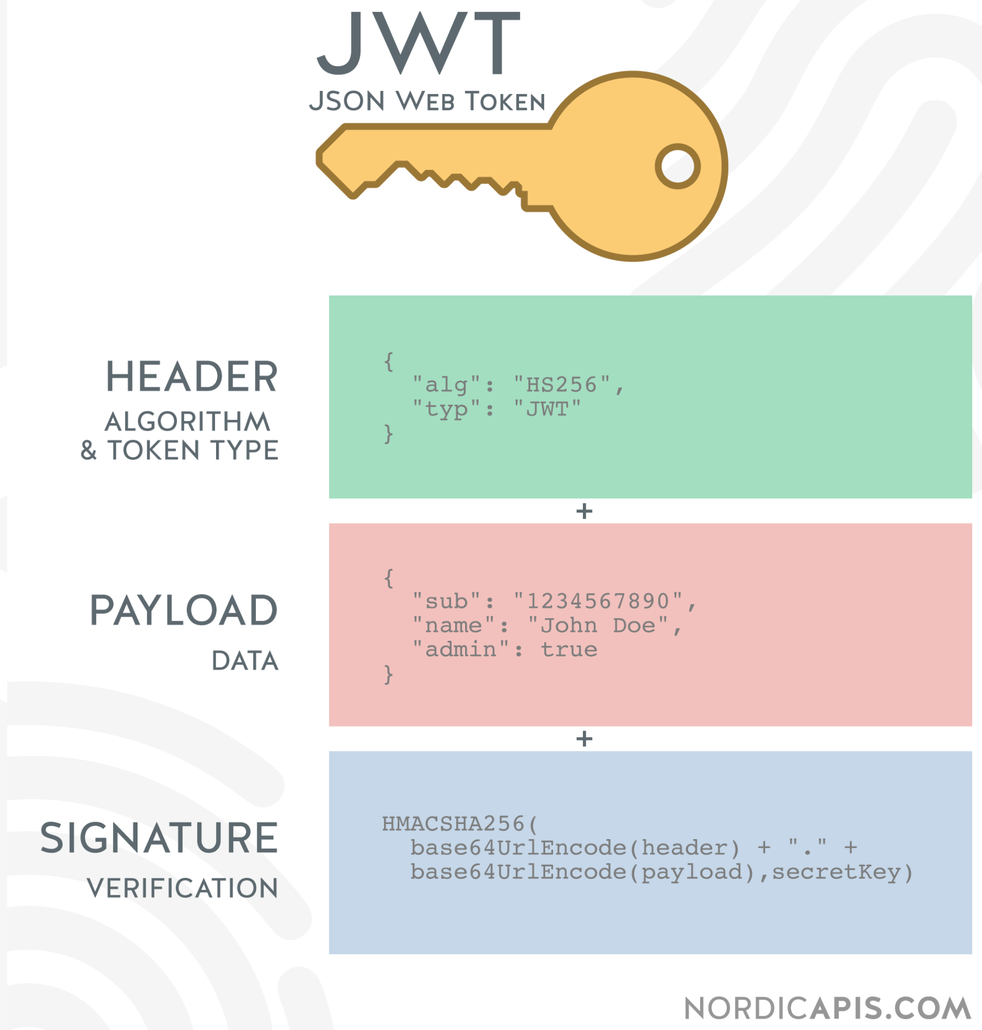
- How does an application “remember” that you were logged in?
- Server either creates a **session**, or gives the client a **token**
  - Session - something the server tracks by using a **cookie**
  - Token - something the server gives the client to pass along in any future requests


Client
Browser
Server
Logs in with username, password
Authenticates
Stores Session as a cookie
Creates session and saves in memory
Requests private user information
Appends cookie data to request
Checks request cookie with stored data in memory
Handles response
Sends response


Client
Browser
Server
Logs in with username, password
Authenticates
Creates a JSON web token with user info, signature and expiry
Saves JSON web token sent from server
Checks token signature to ensure its legit
Adds token to header of subsequent requests
Get user info from token
Handle response

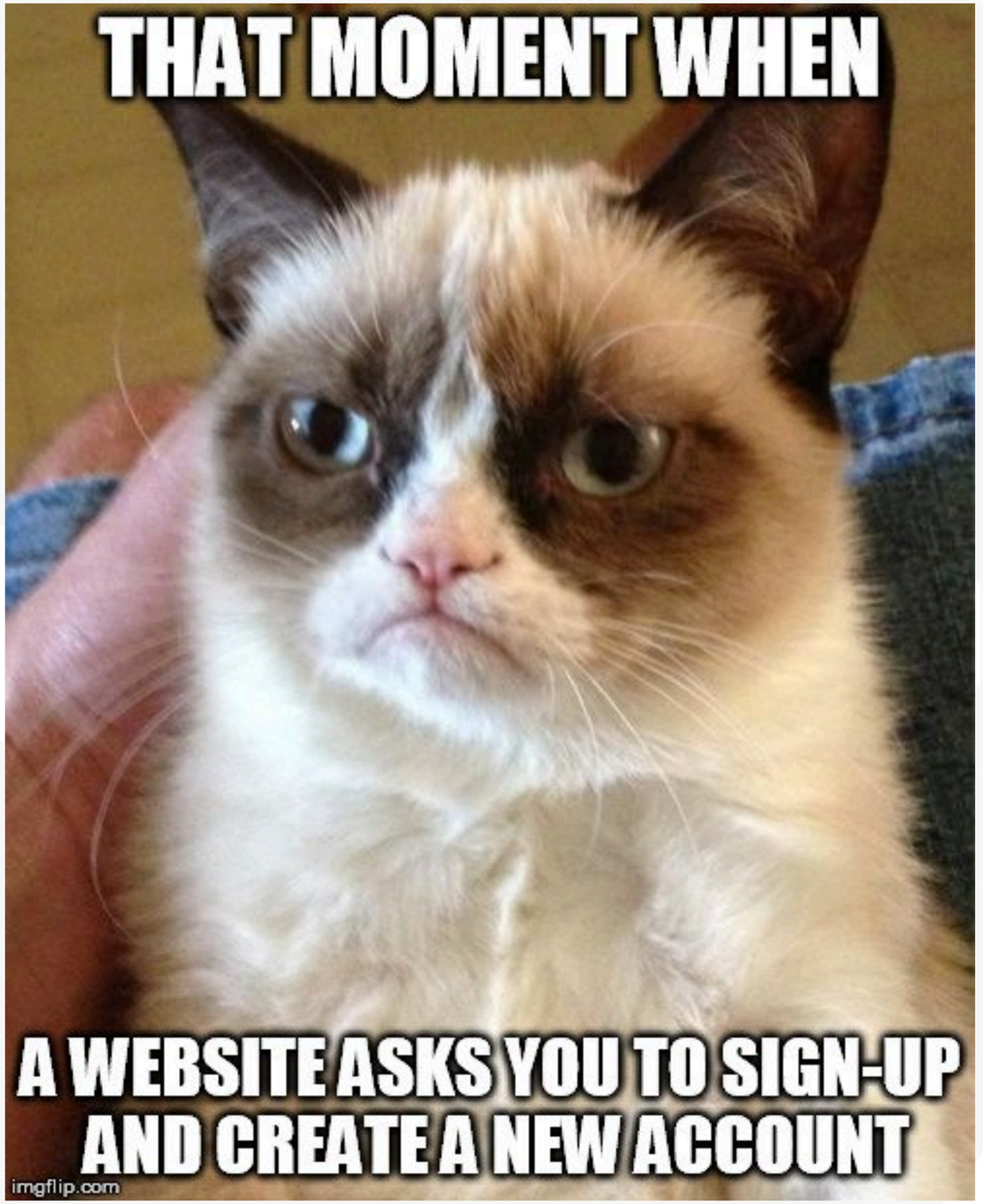
JSON Web Token

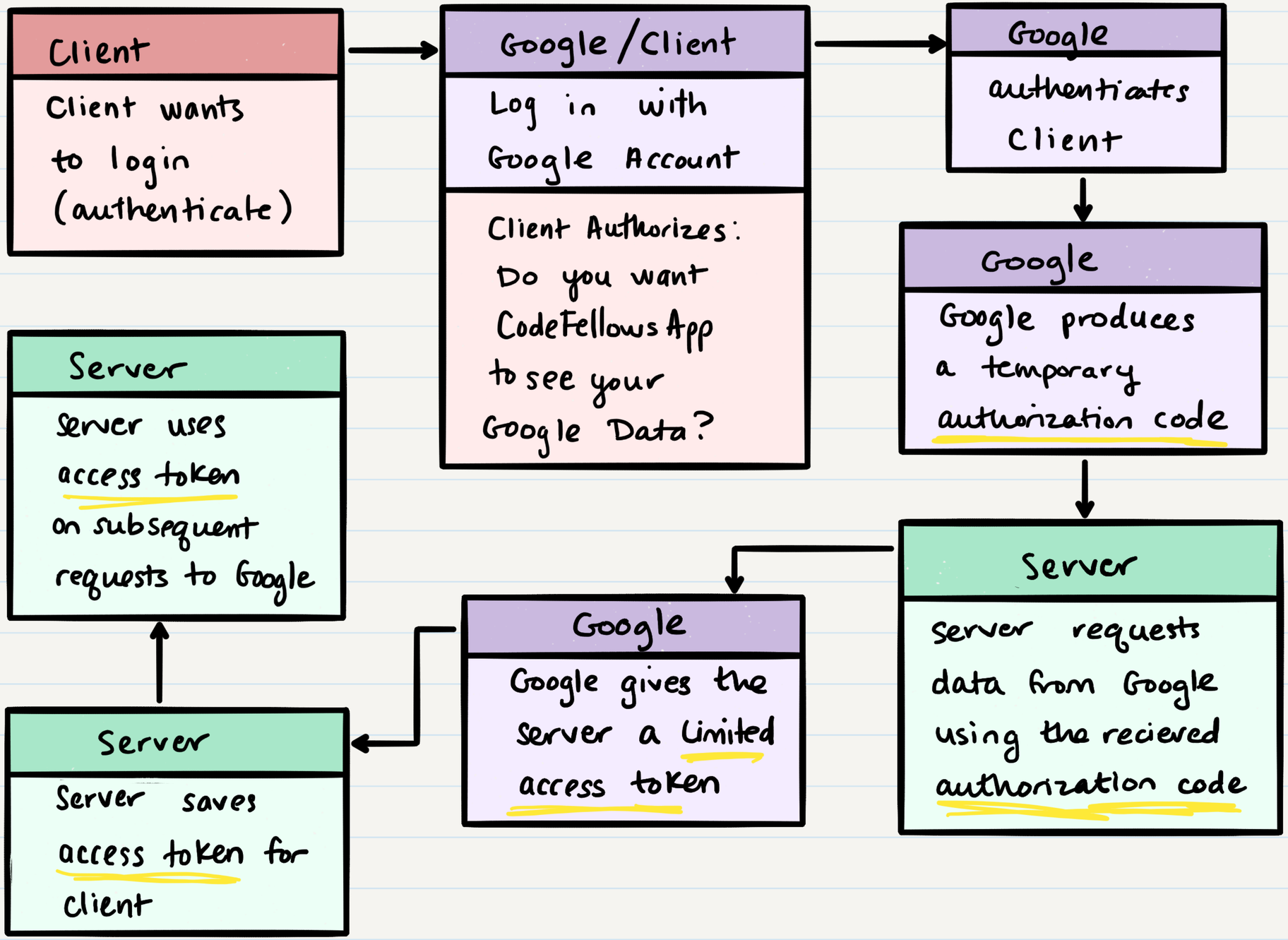
- A very common way to keep a client logged in
- The server generates a unique token
  - This is an encrypted string
  - Only the server has the encryption key
  - The data encrypted is typically some unique reference to the current user (user id for example)
- **JSON Web Token** is a package that generates tokens for us



OAuth

- A standard way for two independent applications to share user data
- Involves a series of “**handshakes**”
- Your application directs the user to a third-party sign in
- Your application gets an **authorization code**
- Your application gets an **access token**
- Your application makes an API request with the access token




Client
Client wants to login (authenticate)
Google / Client
Log in with Google Account
Client Authorizes: Do you want CodeFellows App to see your Google Data?
Google
authentiates Client
Google
Google produces a temporary authorization code
Server
server requests data from Google using the recieved authorization code
Google
Google gives the server a Limited access token
Server
Server saves access token for client
Server
server uses access token on subsequent requests to Google

Lab 12 Overview