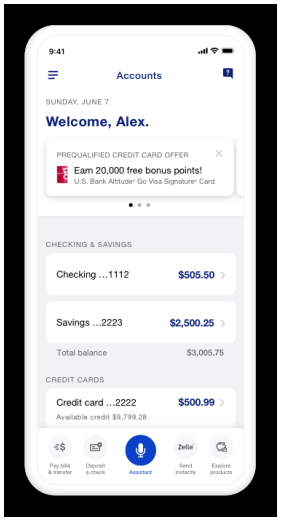
Authentication

- Authentication – who are you?
- Users
- Passwords
- Tokens

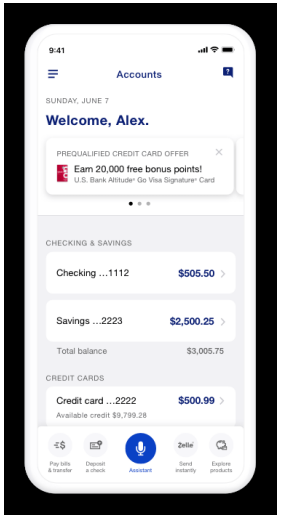
HTTPS "handshake"

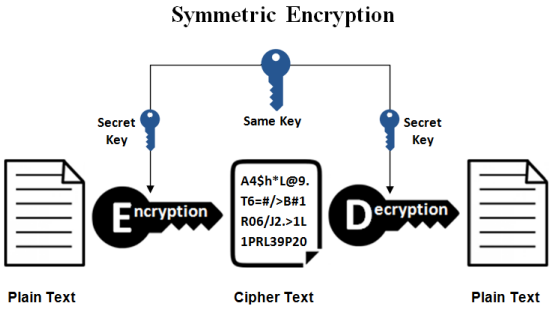
- **HTTPS is designed to prove identity *and* encrypt**



Symmetric encryption

- **After HTTPS handshake, client and server exchange a single key for faster symmetric encryption**
  - *Single key can be used by both parties to encrypt & decrypt*





User authentication

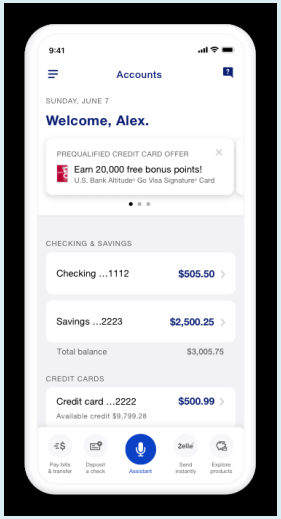
- **Now that we have a secure connection…**
- **The user needs to prove their identify**

API calls

- **Every API call needs to identify the user, and carry proof of their identify…**
  - *otherwise you could access any account you want!*

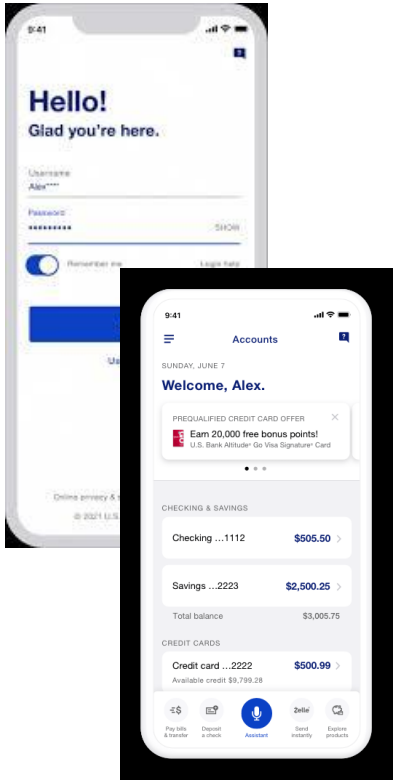
Approach #1

- **Send username and password…**
  - *Requires storing password or prompting each time*



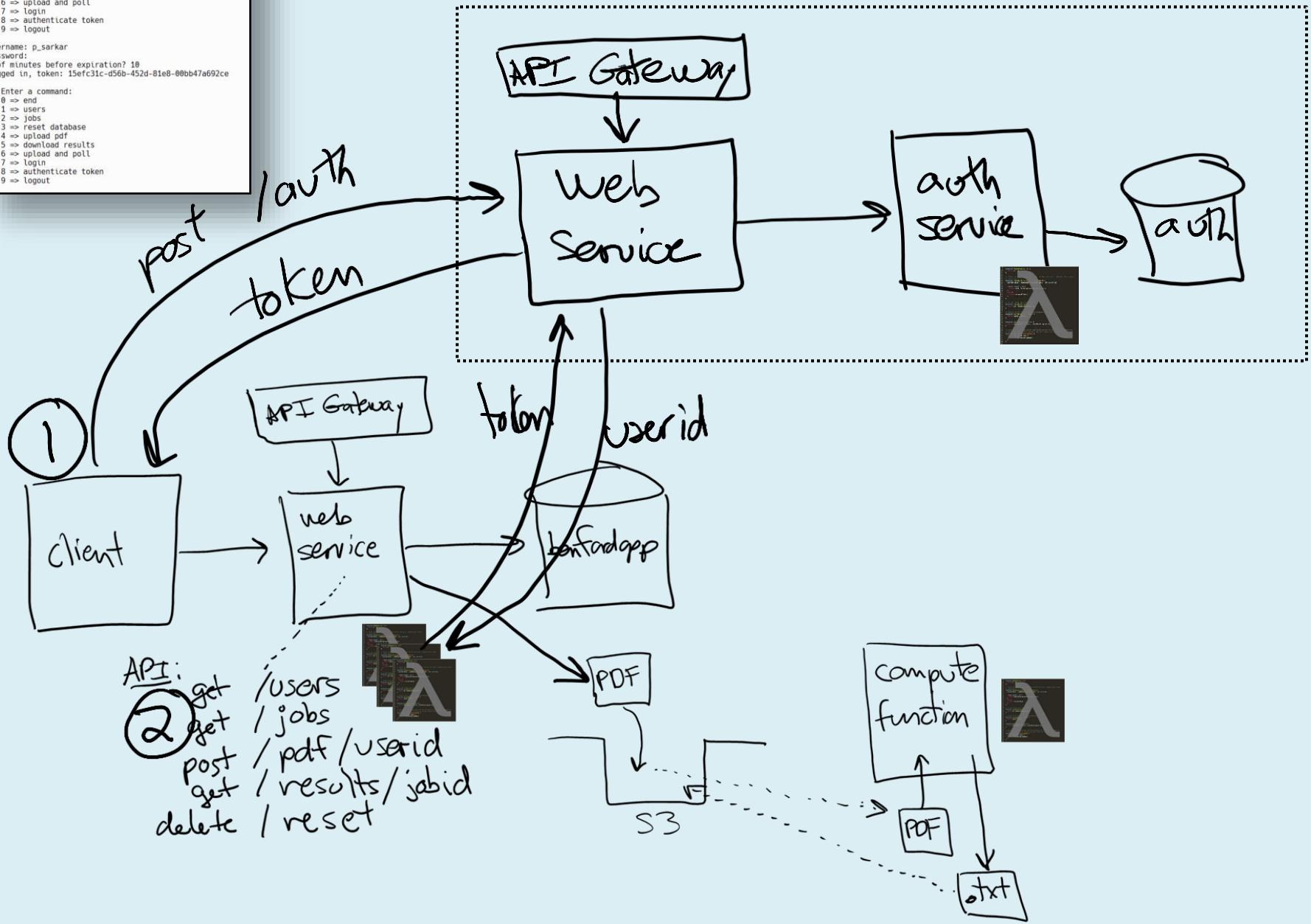
Approach #2

- **Login yields an access token**
  - *Access token is used for identification*
  - *Token can be set to expire after time period (short or long)*



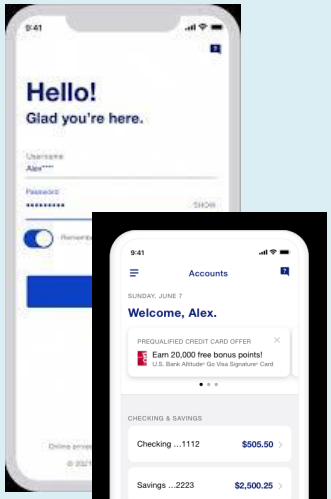
Example: project 04

```
** Welcome to BenfordApp with Authentication **

Config file to use for this session?
Press ENTER to use default, or
enter config file name>


>> Enter a command:
   0 => end
   1 => users
   2 => jobs
   3 => reset database
   4 => upload pdf
   5 => download results
   6 => upload and poll
   7 => login
   8 => authenticate token
   9 => logout
7
username: p_sarkar
Password:
# of minutes before expiration? 10
logged in, token: 15efc31c-d56b-452d-81e8-00bb47a692ce

>> Enter a command:
   0 => end
   1 => users
   2 => jobs
   3 => reset database
   4 => upload pdf
   5 => download results
   6 => upload and poll
   7 => login
   8 => authenticate token
   9 => logout
```



Approach #3

- **Login yields a stateless token**
  - *Token itself contains all necessary info --- nothing on server*
  - *Encrypted using server's private key at login, API functions use public key to validate --- avoids API functions calling auth service*

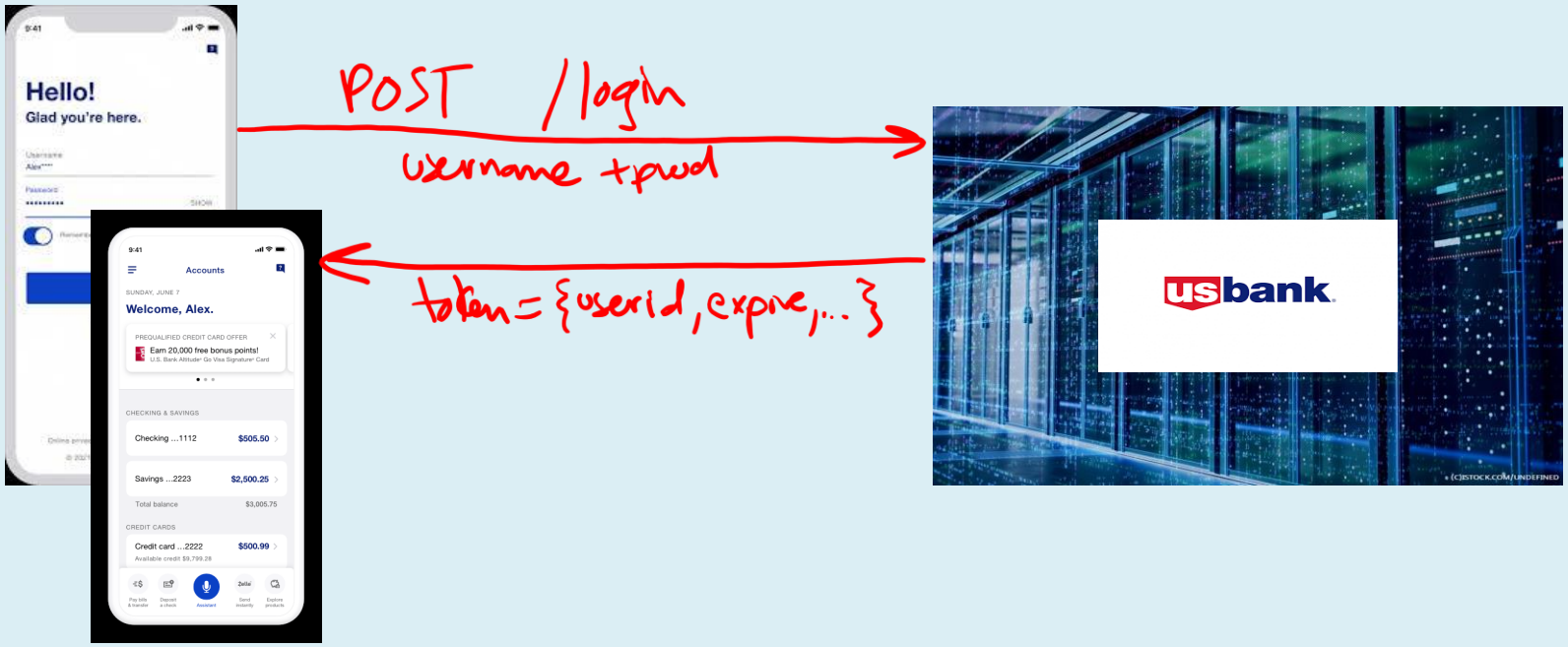


Example:
JSON Web Tokens (JWT)

Question

- **Login yields a stateless token…**
- **Can the client modify the token? Change the userid or expiration?**

No! But why not?



That's it, thank you!