Identity & Security

From Authentication to Auditing

About Me

- Brian Pontarelli
- Colorado Native
- CEO of Inversoft
- Been coding for 22 years
- OWASP member
- Worked at XOR, BEA, Orbitz, and others
- Author (various articles and an eBook)
- Father of 3
- Drummer
- Home brewer

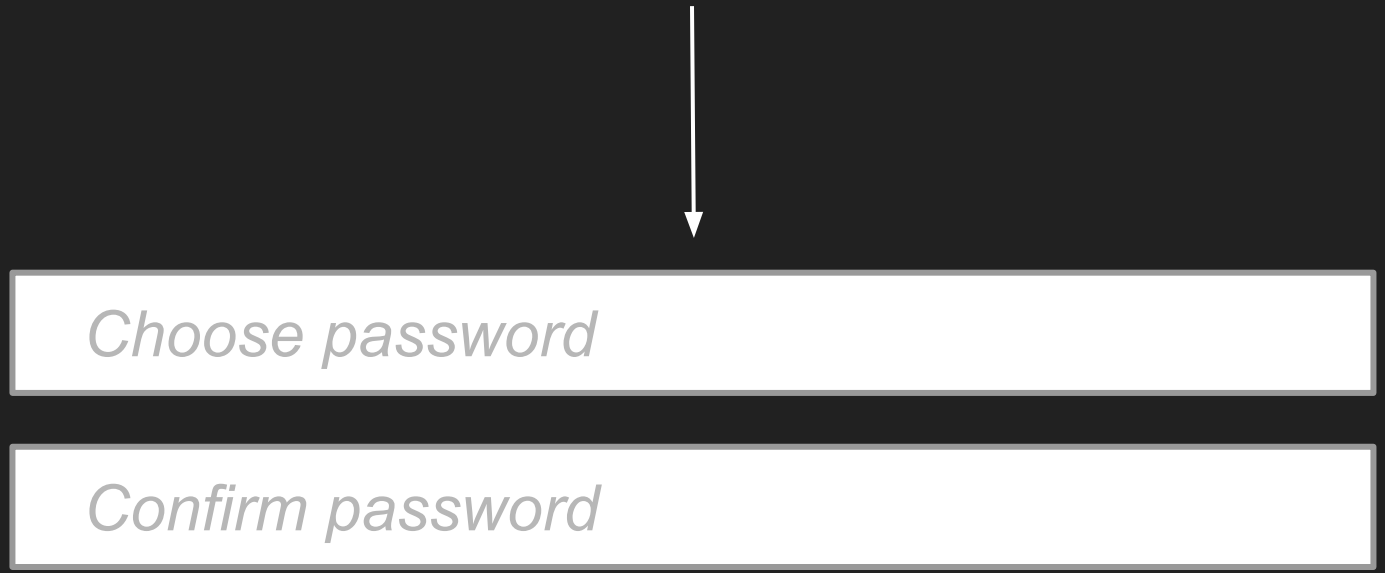
Account Creation

Two Models

- Self registration
  - You create your own account
- Account provisioning
  - Someone else creates your account
  - Please never email passwords plain-text

Provisioning Accounts

```
https://www.inversoft.com/setup-account?id=SEK3lkau9lkjAJLKljk309
uljkasdklfhlsdg293jKLJajlklkjend
```



Password Security

Large bitcoin rigs can do 1,000 Tera-hashes per second.

Assuming passwords normally contain up to 100 characters…

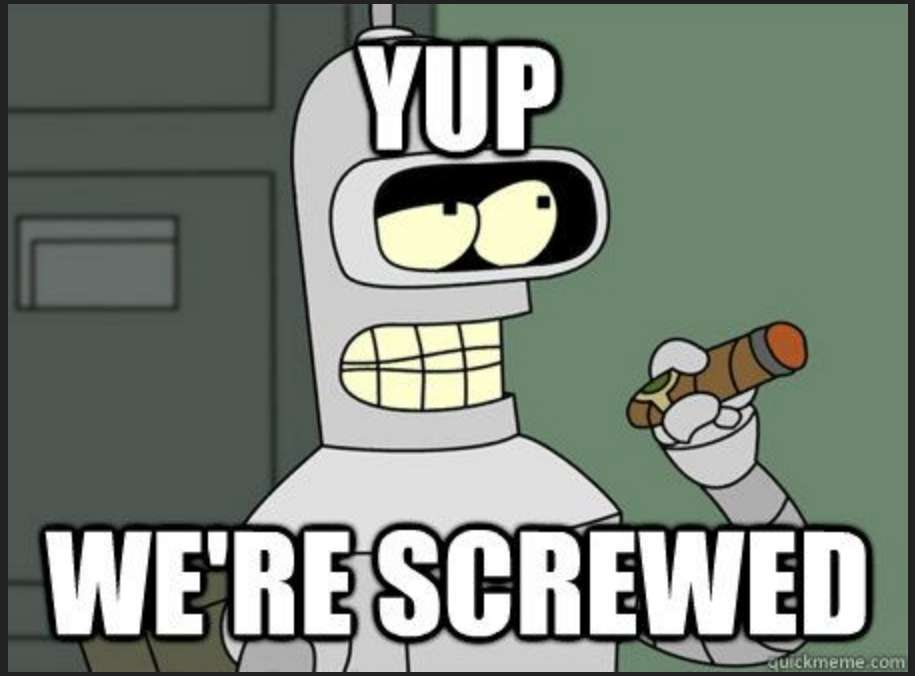
That’s 1e18 possible passwords

(for up to 9 character passwords)

Generating every possible hash for these passwords would take 1e3 seconds

Also known as 16 minutes

If you stop at 8 characters, it can take less than one minute


YUP
WE'RE SCREWED
quickmeme.com

Complexity

- Encryption is all about complexity
- Slow algorithms mean good security
- SHA and MD5 have iterations, Bcrypt and others have load factor
- More iterations = GOOD
- More load = GOOD


36% COMPLETE

A Few Common Algorithms

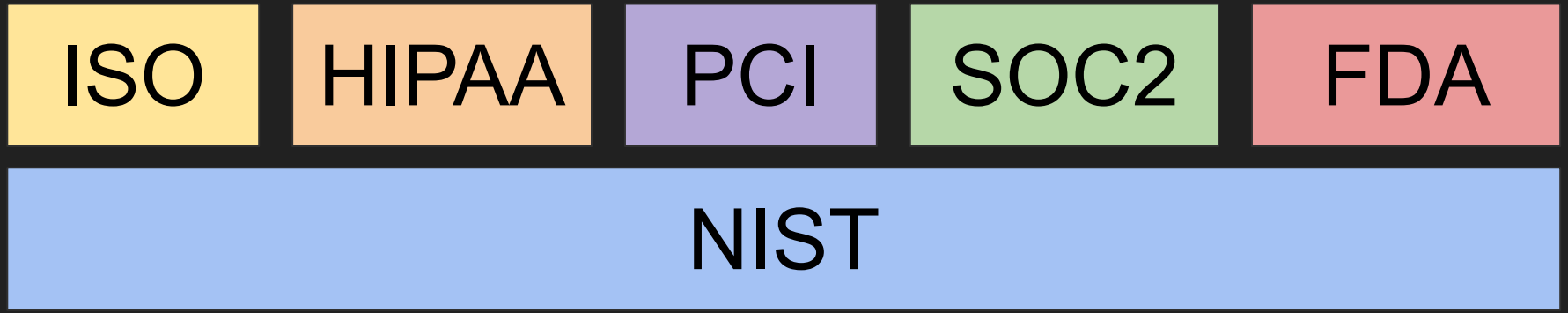
- PBKDF2
- BCrypt
- SHA
- MD5
- SCrypt

Salting

- Add some junk before the password
- Then hash that!
- Prevents table lookups

```
[{16e49f4ffd8741b9801357ed3b0403d8}]password
```

Password Rules


ISO
HIPAA
PCI
SOC2
FDA
NIST

NIST

- NIST provides the base rules
- Everyone else eventually updates based on NIST
- This can take 3-5 years
- NIST just released a bunch of changes

Comparison

NIST

- 8 characters minimum
- Maximums should be at least 64 or have no limit
- No mandatory changes
- No special character requirements

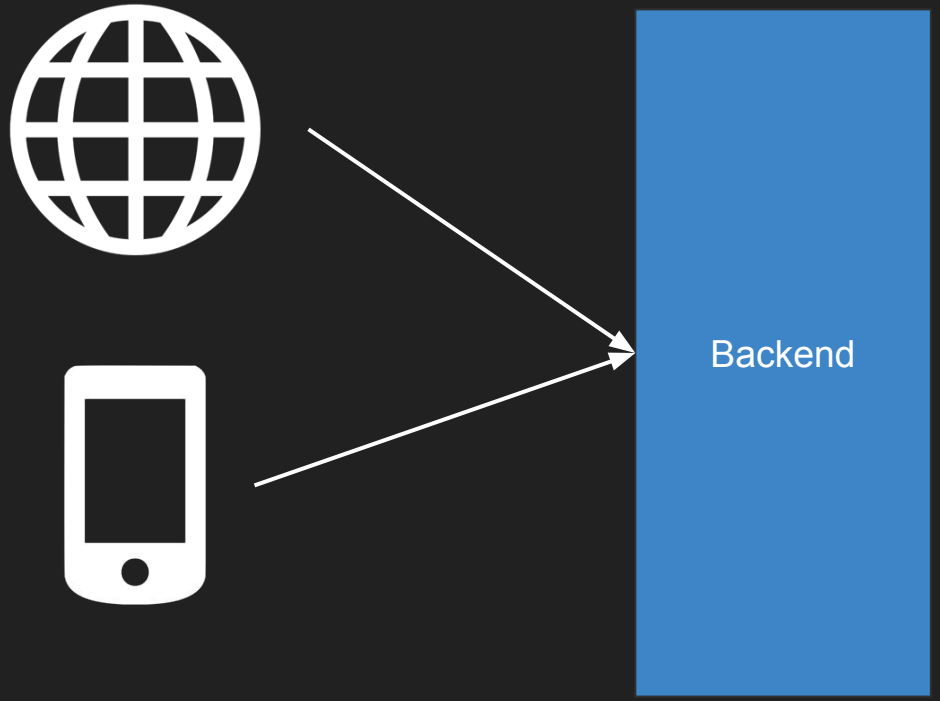
HIPAA

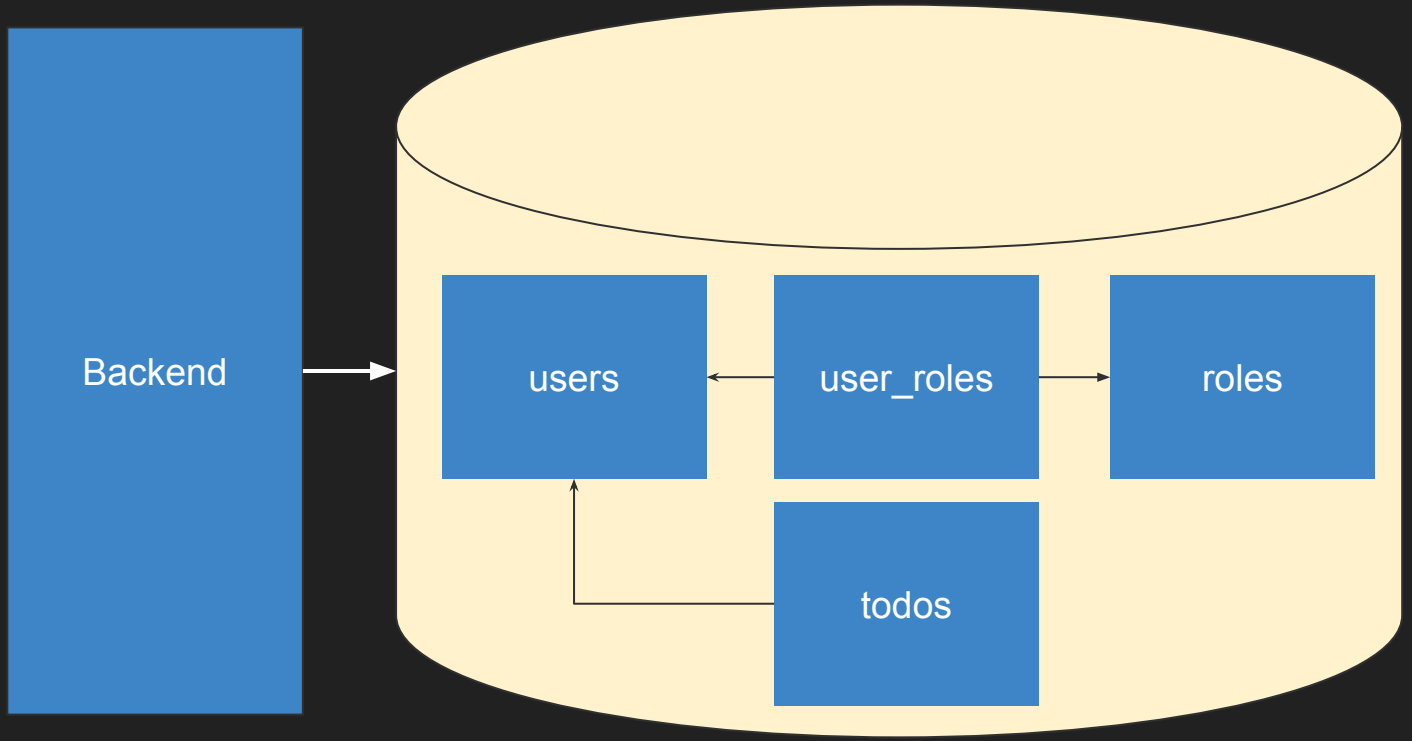
- At least 6 characters (8 preferred)
- No more than 32 characters
- Uppercase, lowercase, numbers and symbols required
- Passwords must be changed every 45 to 90 days
- Can’t reuse the last 12 passwords

Authentication

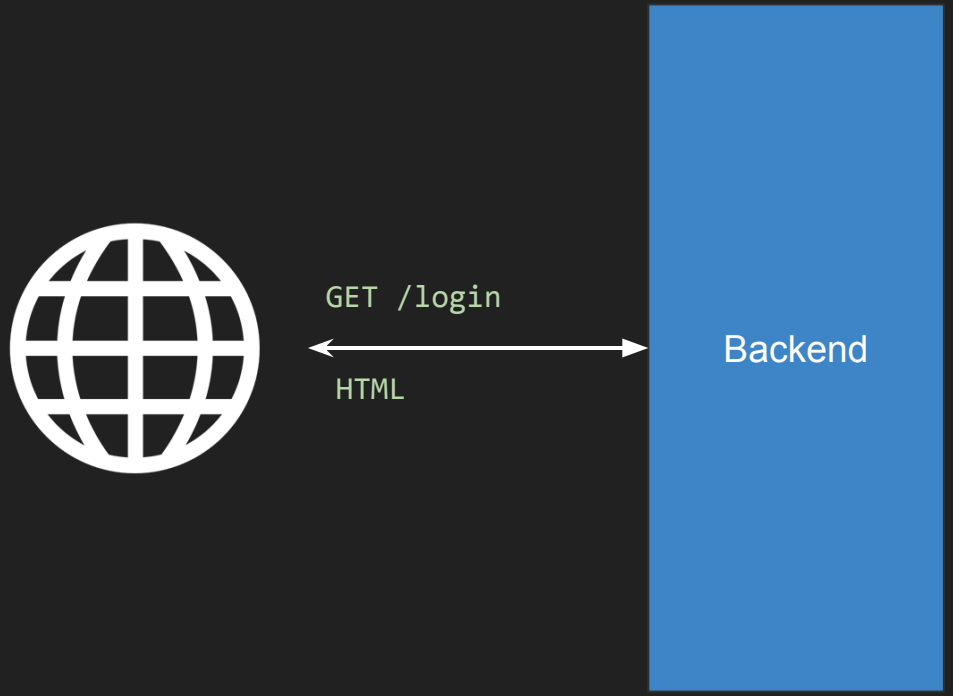
brian@inversoft.com

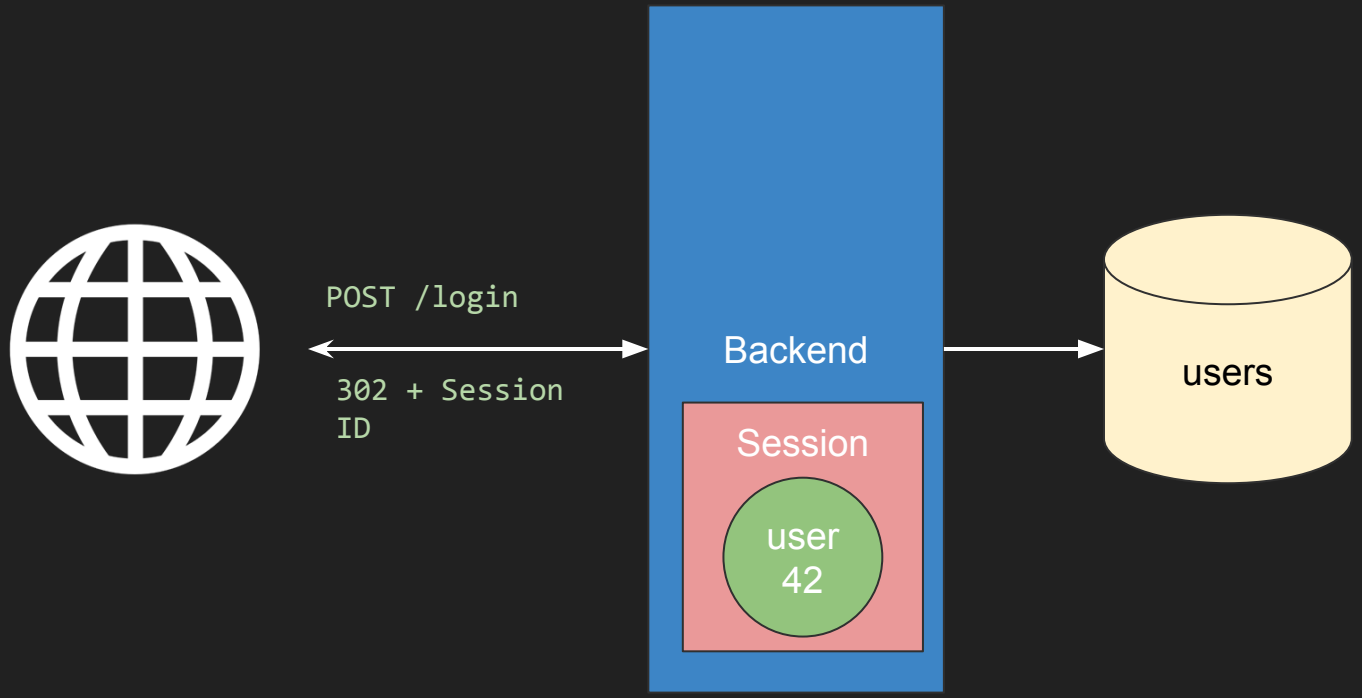
****

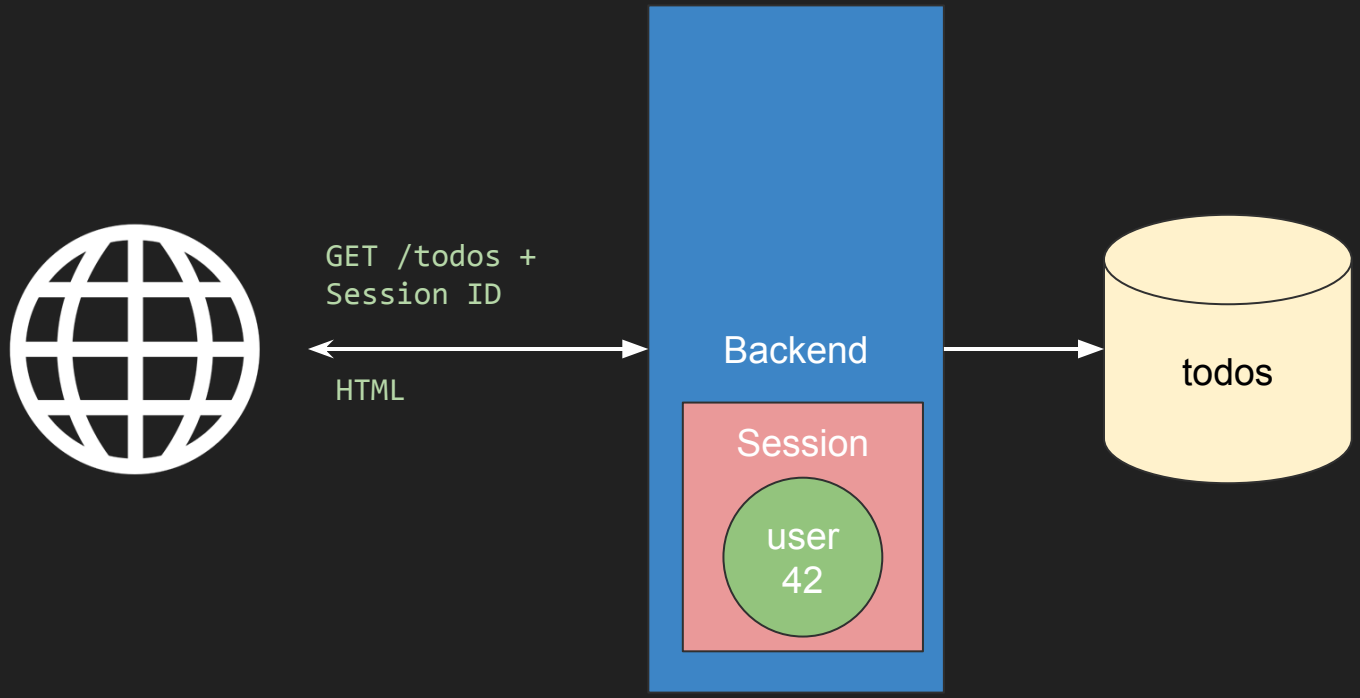

Backend


Backend
users
user_roles
roles
todos

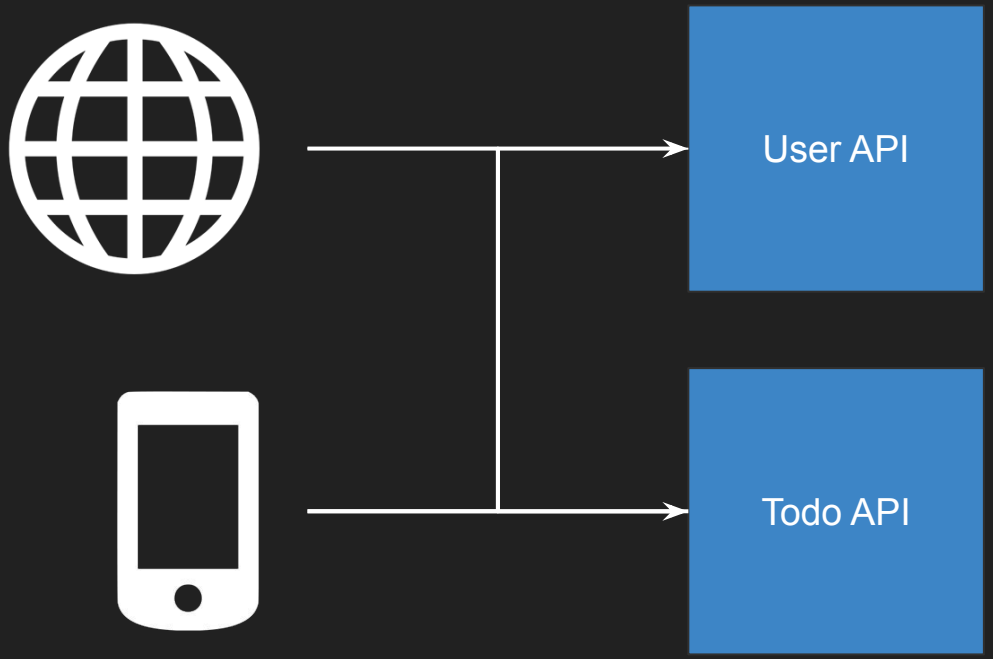
```
CREATE TABLE todos (
  id INT NOT NULL,
  text TEXT NOT NULL,
  user_id INT NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT todos_fk_1 FOREIGN KEY (user_id)
    REFERENCES users(id) ON DELETE CASCADE
);
```

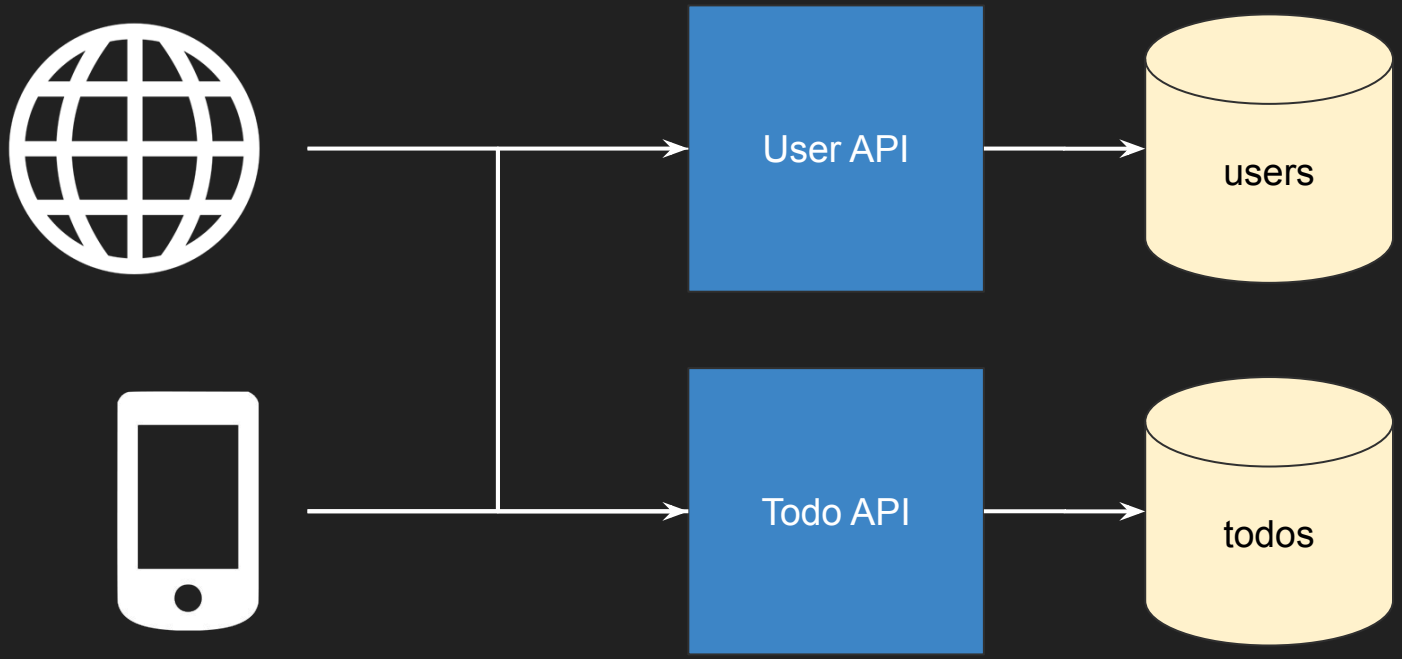

GET /login
HTML
Backend

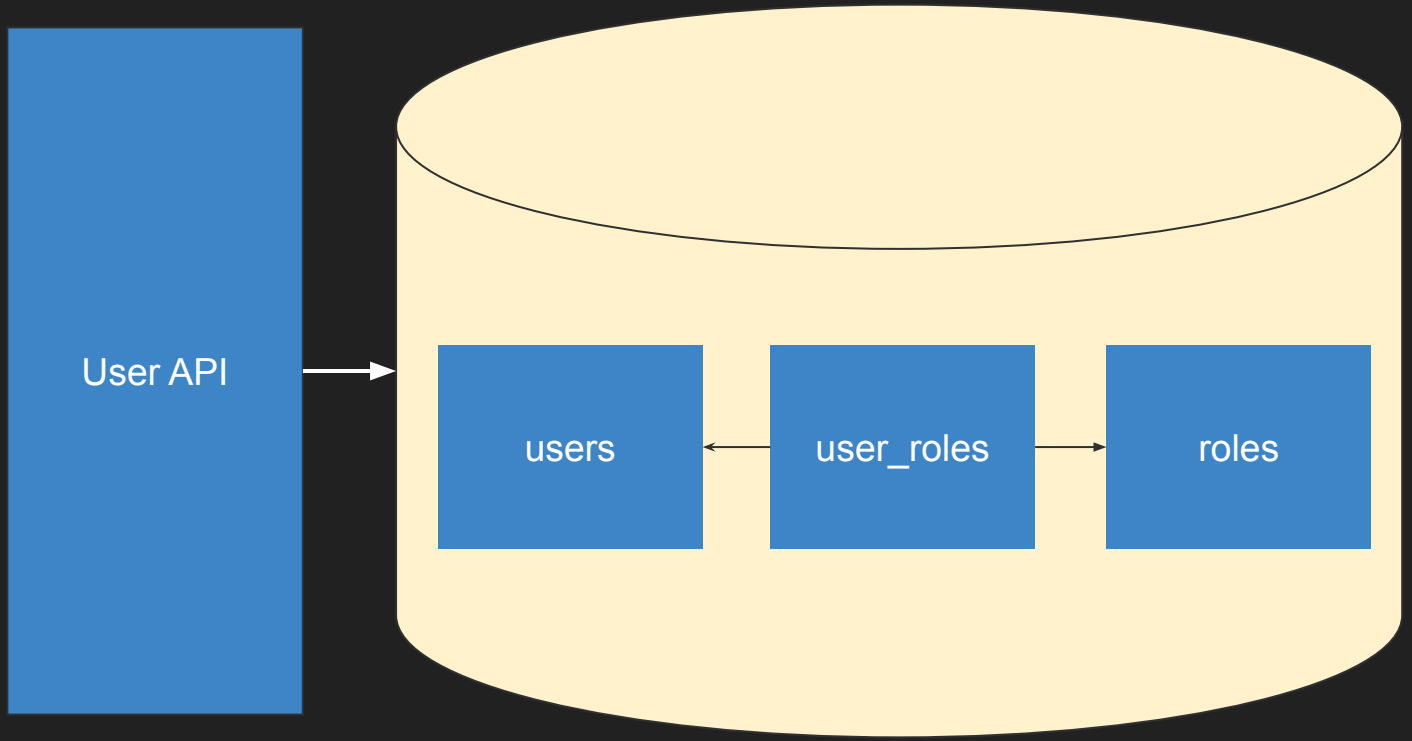

POST /login
302 + Session ID
Backend
Session
user 42
users

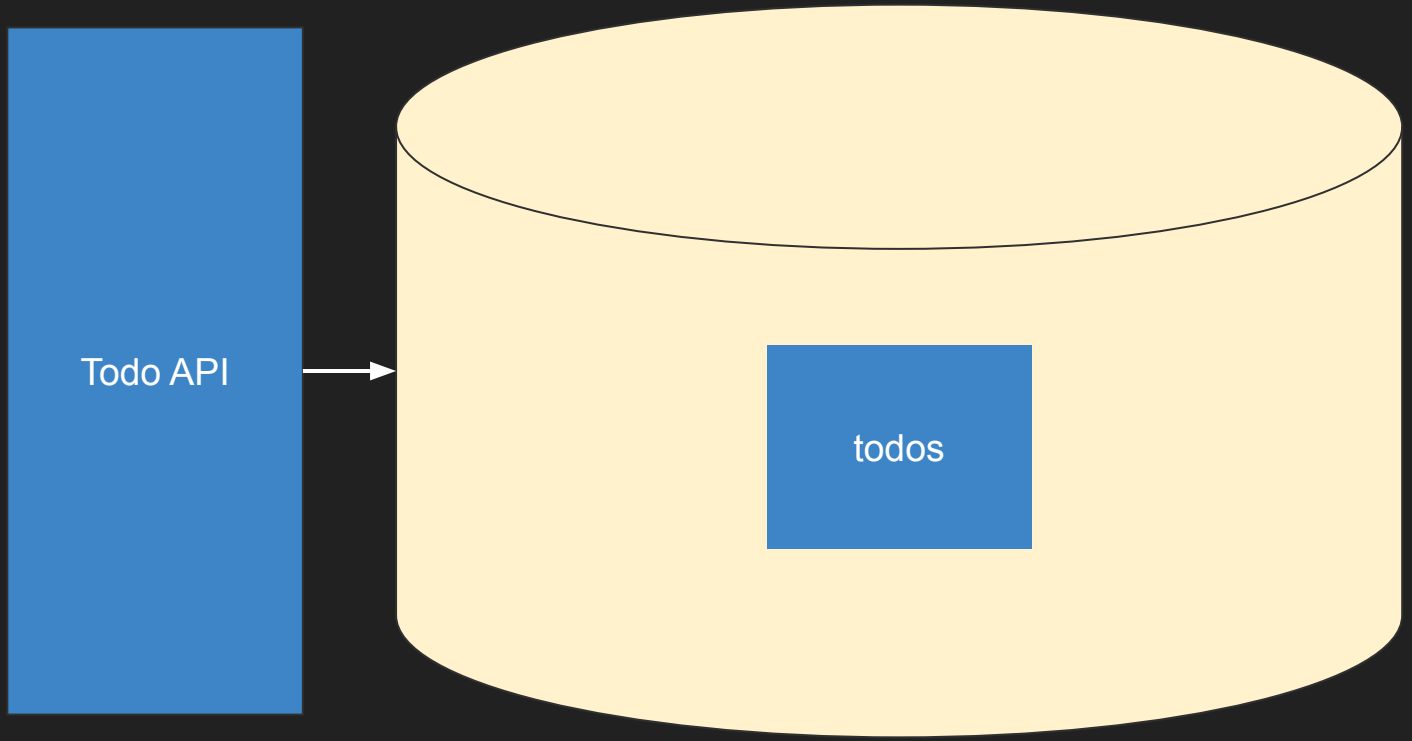

GET /todos +
Session ID
HTML
Backend
Session
user
42
todos

```sql
SELECT * FROM todos WHERE user_id = 42;
```

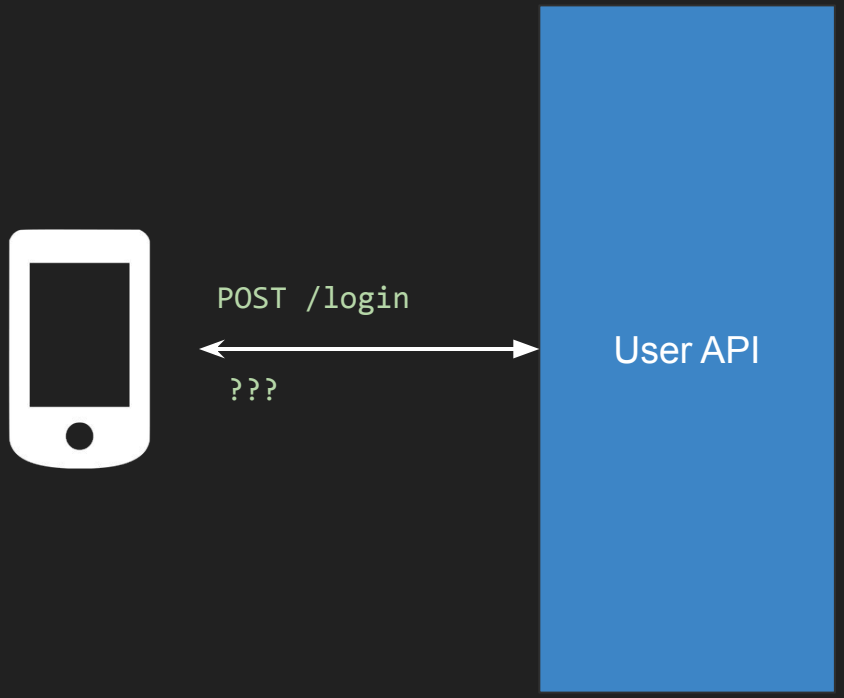

User API
Todo API

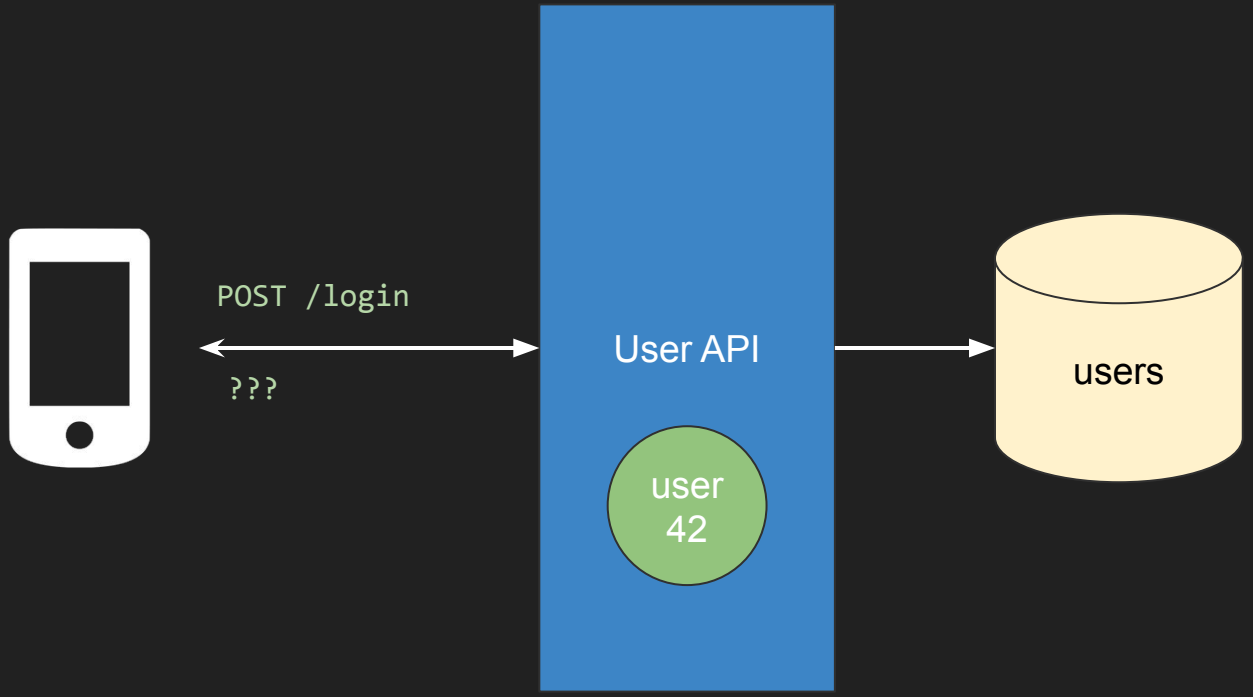

User API
users
Todo API
todos

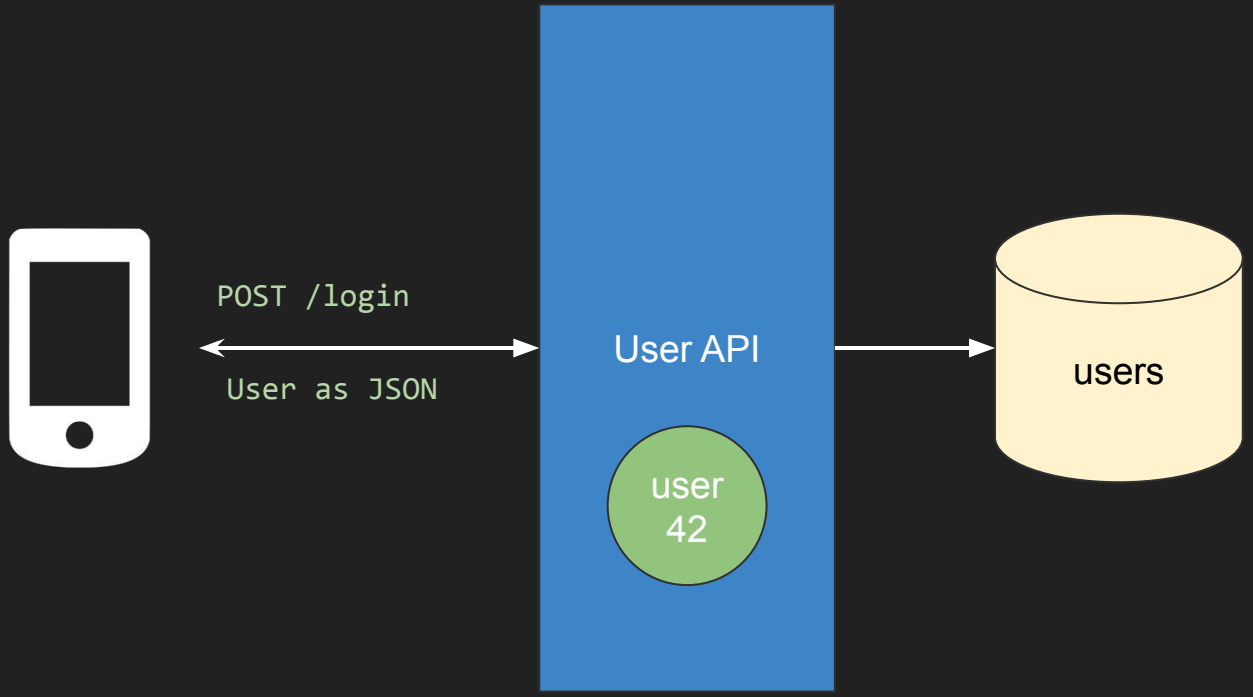

User API
users
user_roles
roles


Todo API
todos

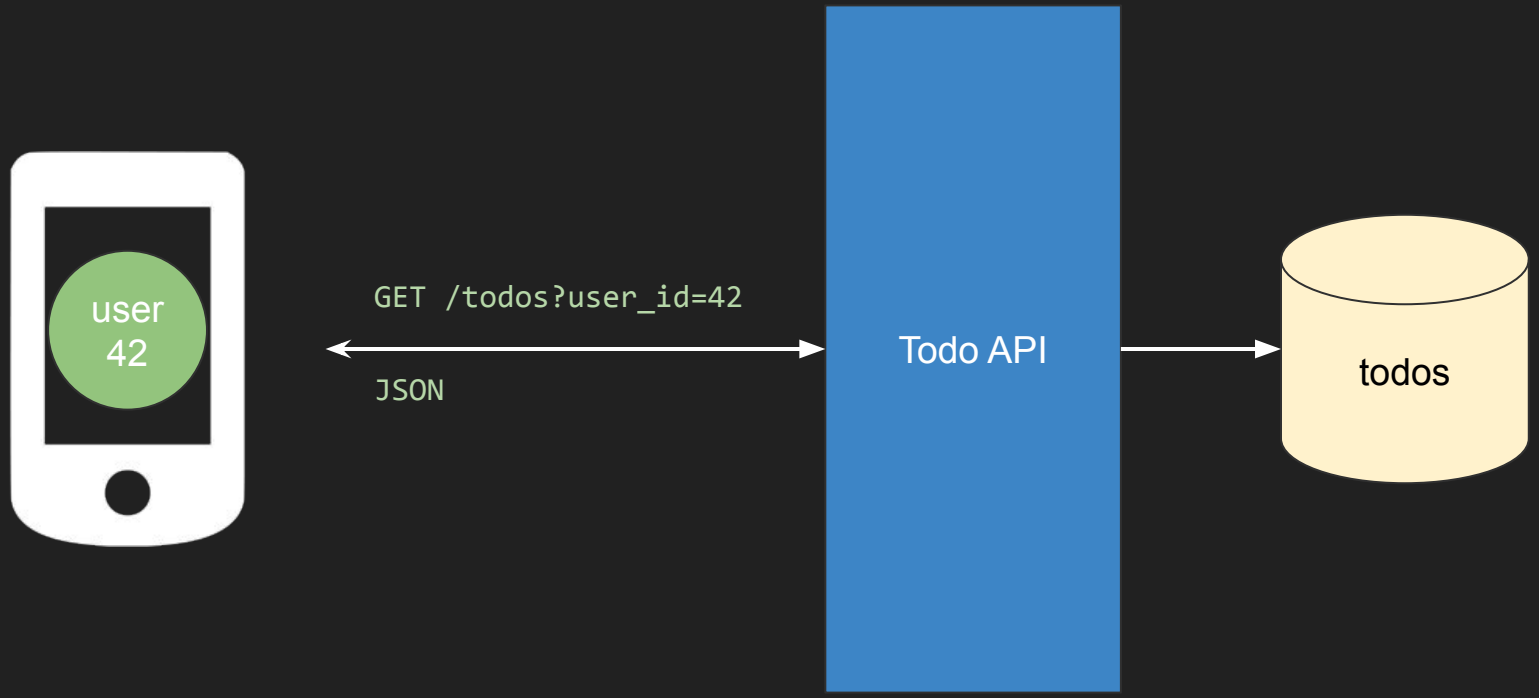
```
CREATE TABLE todos (
  id INT NOT NULL,
  text TEXT NOT NULL,
  user_id INT NOT NULL,
  PRIMARY KEY (id)
);
```


POST /login
???
User API

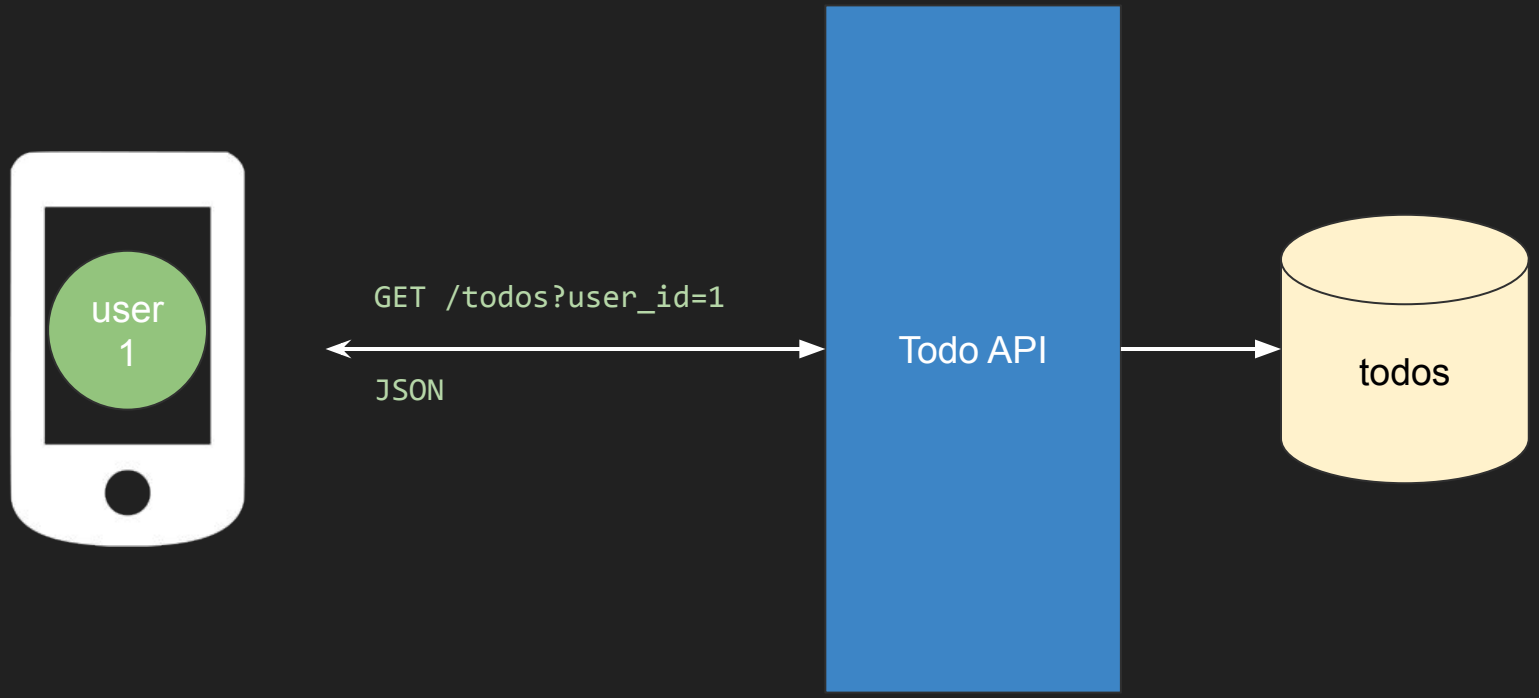

POST /login
???
User API
user
42
users

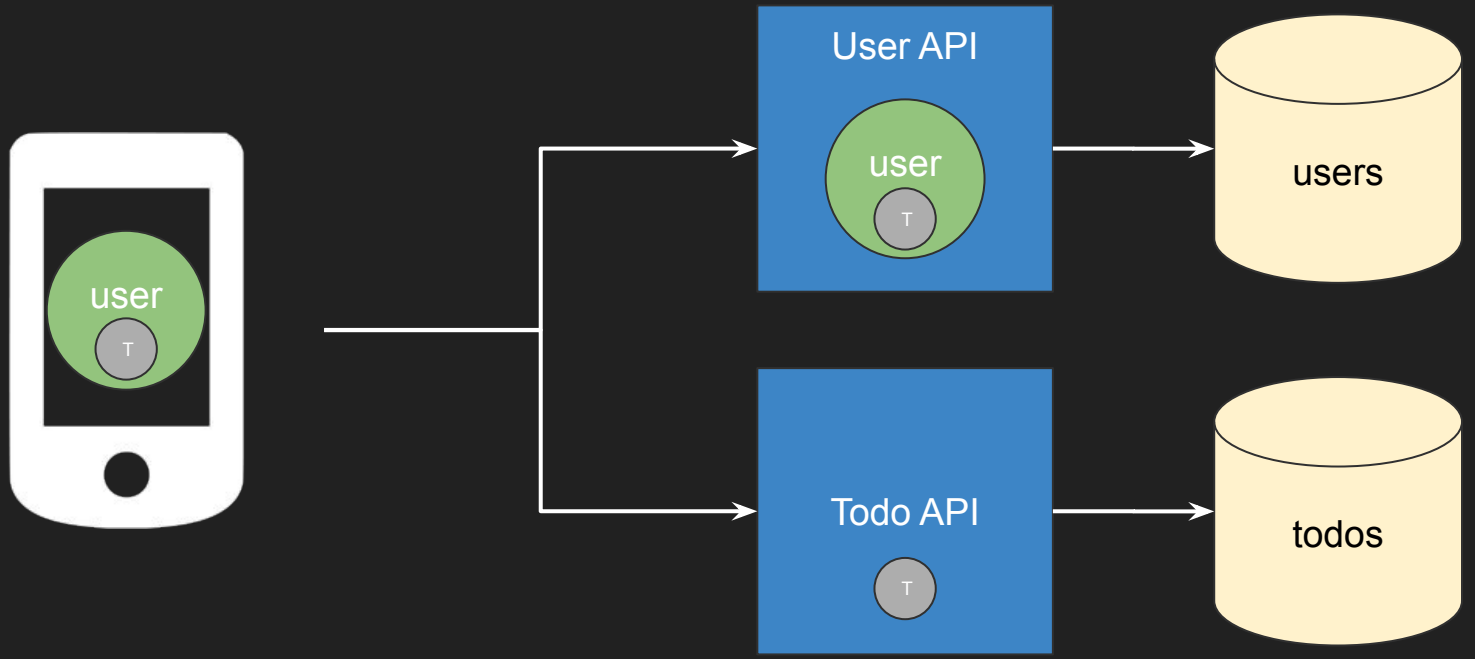

POST /login
User as JSON
User API
user
42
users

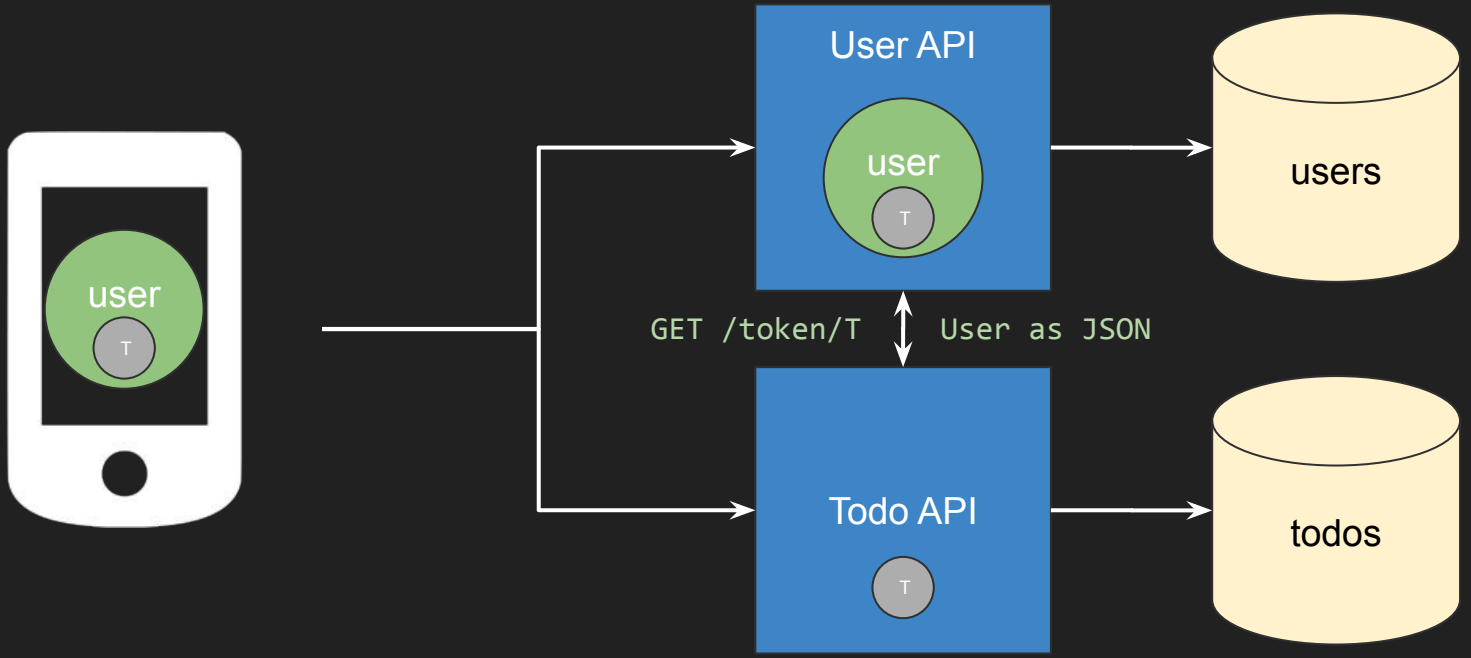
```
{
  "user": {
    "id": 42,
    "name": "Brian Pontarelli",
    "email": "brian@inversoft.com",
    "roles": ["admin"]
  }
}
```


user
42
GET /todos?user_id=42
JSON
Todo API
todos

Danger Will Robinson!

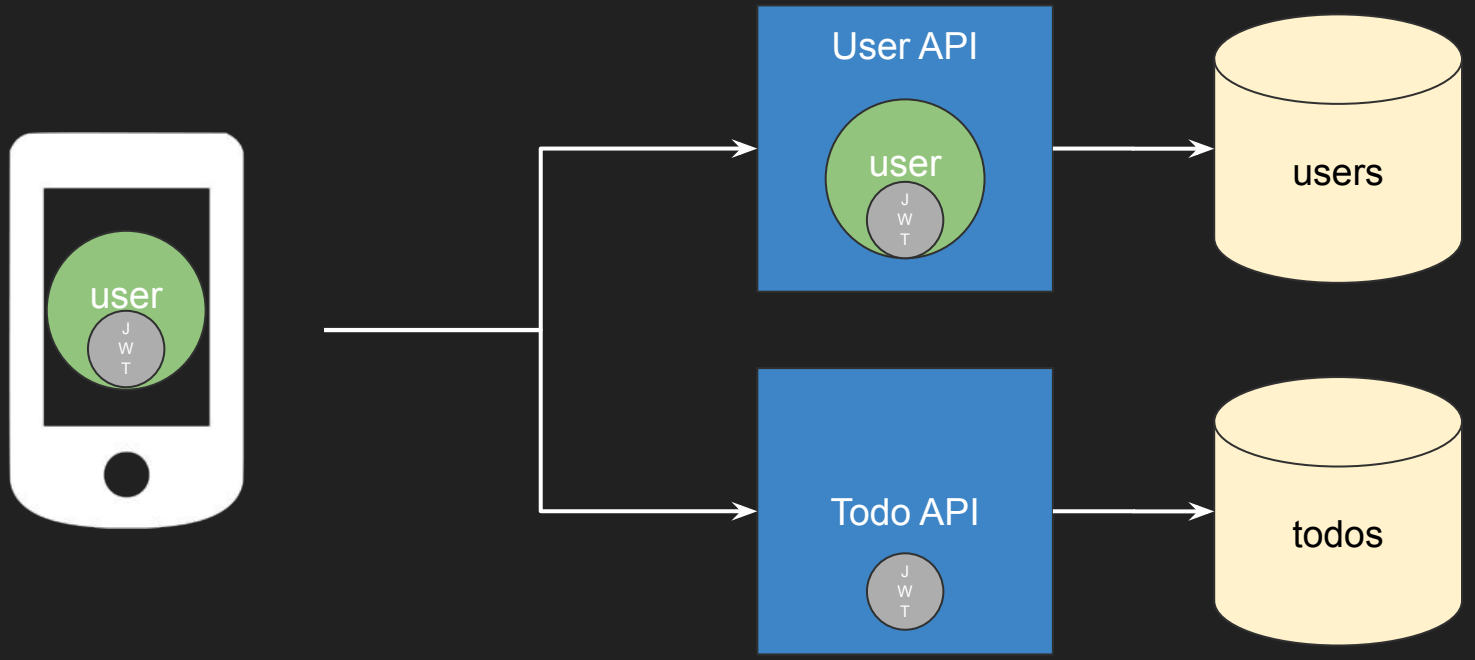

user
1
GET /todos?user_id=1
JSON
Todo API
todos


user
T
User API
user
T
Todo API
T
users
todos


user
T
User API
user
T
users
GET /token/T
User as JSON
Todo API
T
todos

Tokens

- There must be a User -> Token mapping
- In memory or in database
- Makes the User API slightly stateful
- Can be very chatty
- Couples the User API to EVERYTHING (almost)


user
J
W
T
User API
user
J
W
T
users
Todo API
J
W
T
todos

JWT

- JSON Web Tokens
- Signed with a public/private key pair
- Can be validated by Todo API without calling User API
- Contains roles

JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE0ODUxNDA5ODQsImlhdCI6MTQ4NTEzNzM4NCwiaXNzIjoiYWNtZS5jb20iLCJzdWIiOiIyOWFjMGMxOC0wYjRhLTQyY2YtODJmYy0wM2Q1NzAzMThhMWQiLCJhcHBsaWNhdGlvbklkIjoiNzkxMDM3MzQtOTdhYi00ZDFhLWFmMzctZTAwNmQwNWQyOTUyIiwicm9sZXMiOltdfQ.Mp0Pcwsz5VECK11Kf2ZZNF_SMKu5CgBeLN9ZOP04kZo

JWT Header

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
```

=

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

JWT Body

eyJleHAiOjE0ODUxNDA5ODQsImlhdCI6MTQ4NTEzNzM4NCwiaXNzIjoiYWNtZS5jb20iLCJzdWIiOiIyOWFjMGMxOC0wYjRhLTQyY2YtODJmYy0wM2Q1NzAzMThhMWQiLCJhcHBsaWNhdGlvbklkIjoiNzkxMDM3MzQtOTdhYi00ZDFhLWFmMzctZTAwNmQwNWQyOTUyIiwicm9sZXMiOltdfQ

=

```
{
  "iss": "inversoft.io",
  "exp": 1300819380,
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Brian Pontarelli",
  "roles": ["RETRIEVE_TODOS"]
}
```

JWT Signature

`Mp0Pcwsz5VECK11Kf2ZZNF_SMKu5CgBeLN9ZOP04kZo`

=

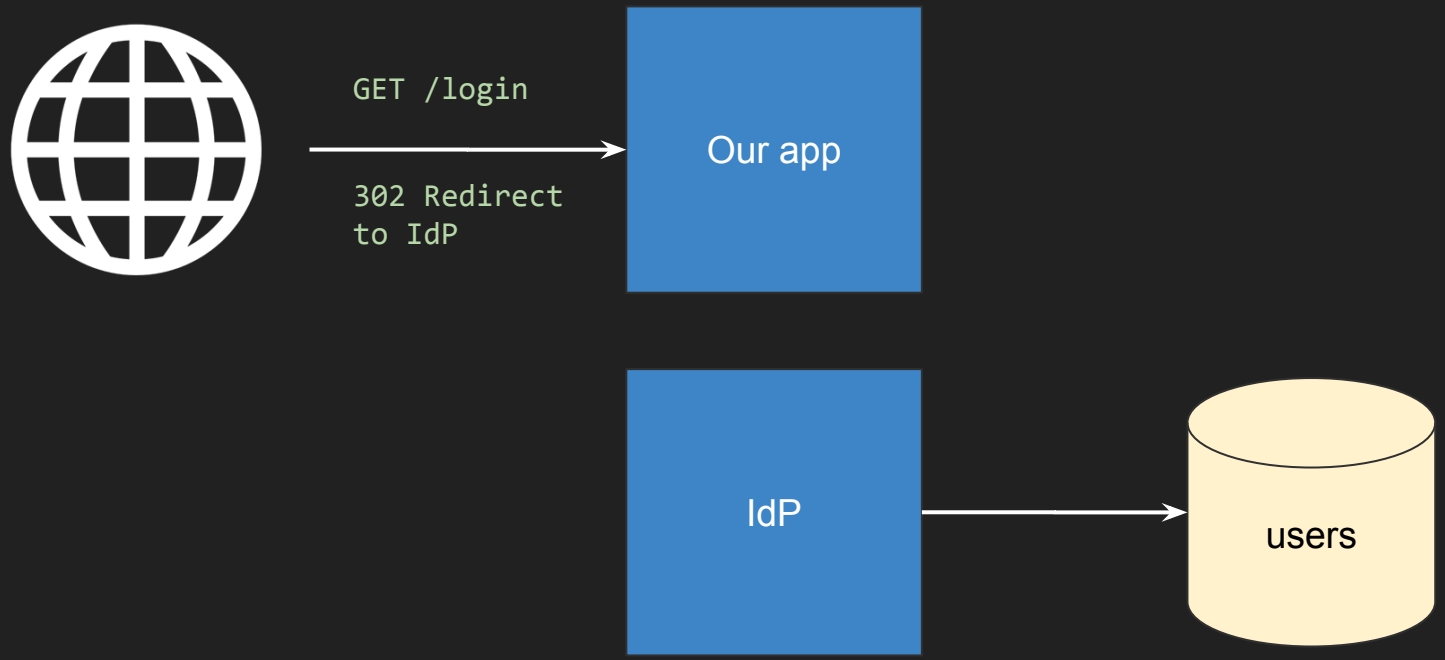
RSA/HMAC Signature

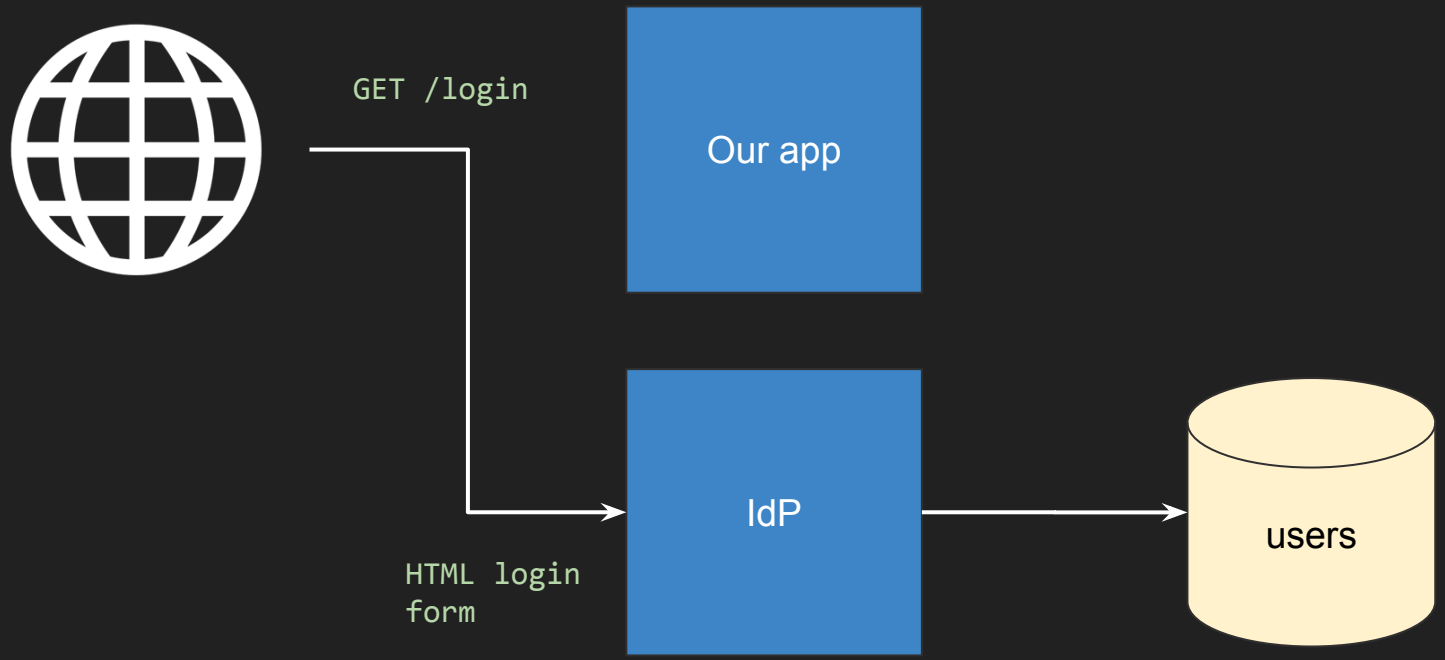
```
select * from todos where user_id =
‘19016b73-3ffa-4b26-80d8-aa9287738677’;
```

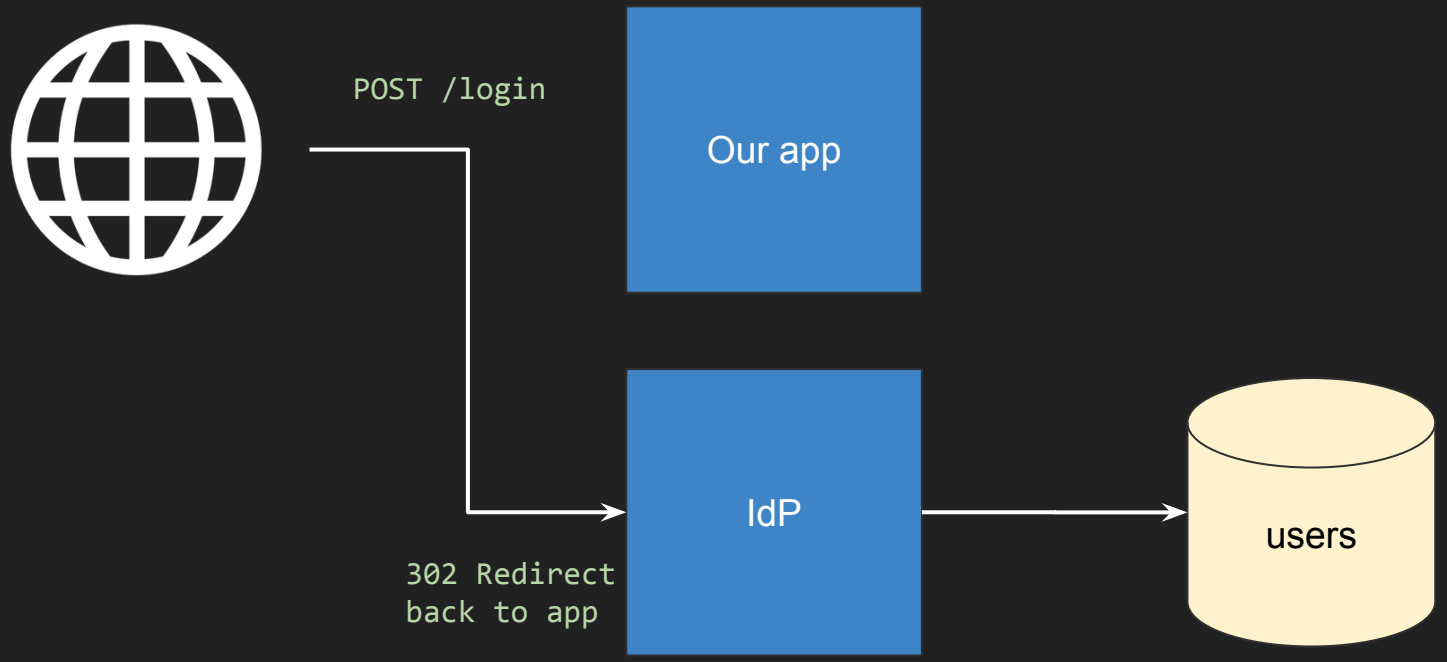
OAuth/SAML
(SSO or Federation)

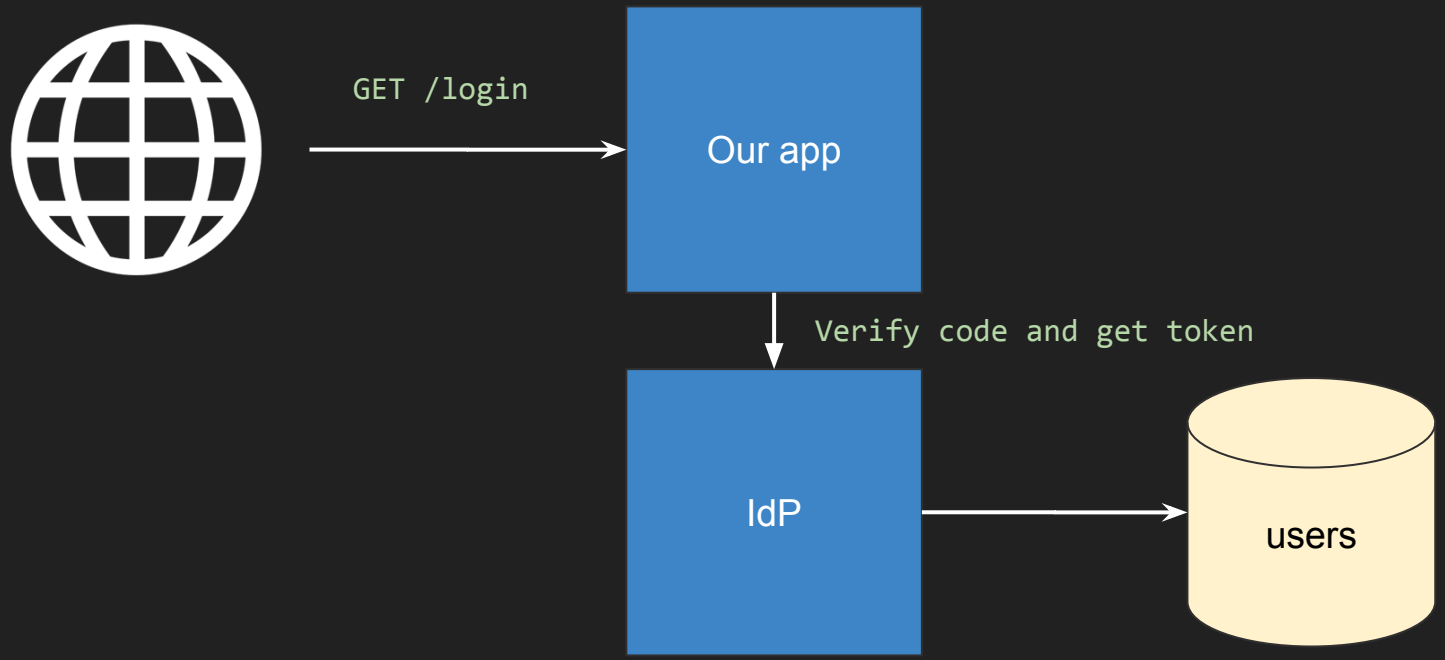
Federated Identity

- Go login over there
- Come back with proof
- I’ll ask that other system if you are legit
- I won’t care about managing passwords
- I might not care about managing user data


GET /login
302 Redirect
to IdP
Our app
IdP
users


GET /login
Our app
IdP
users
HTML login
form


POST /login
Our app
IdP
users
302 Redirect
back to app


GET /login
Our app
Verify code and get token
IdP
users

Most tokens are JWTs now

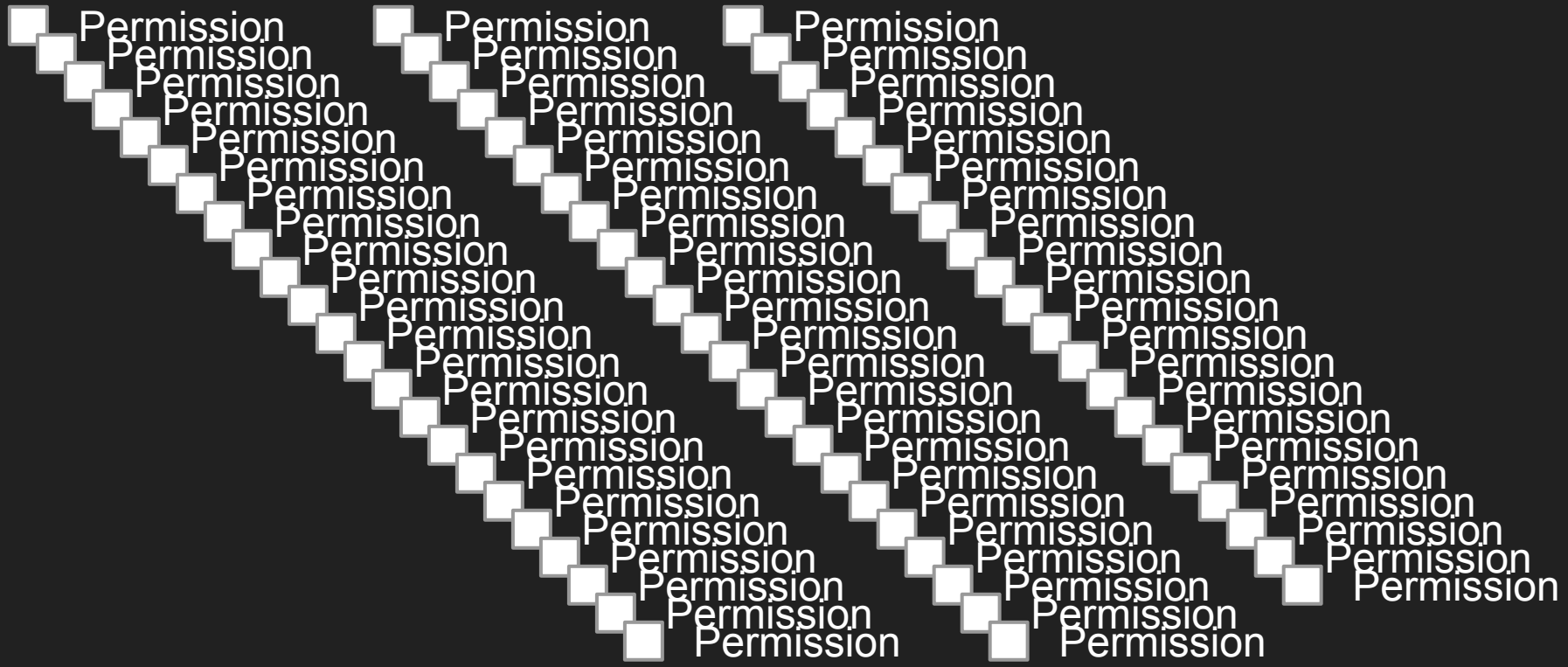
Authorization

- Should this button be visible?
- Who can click it?
- Does it do different things depending on who clicks it?

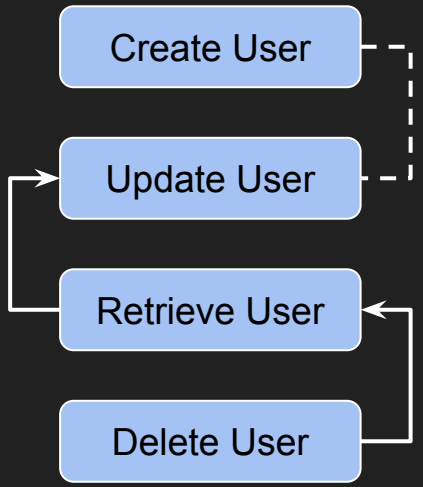
Role Based Access Control

- Also known as RBAC
- Declarative coarse-grained roles
- Roles control permissions
- Permissions can be programmatic or declarative

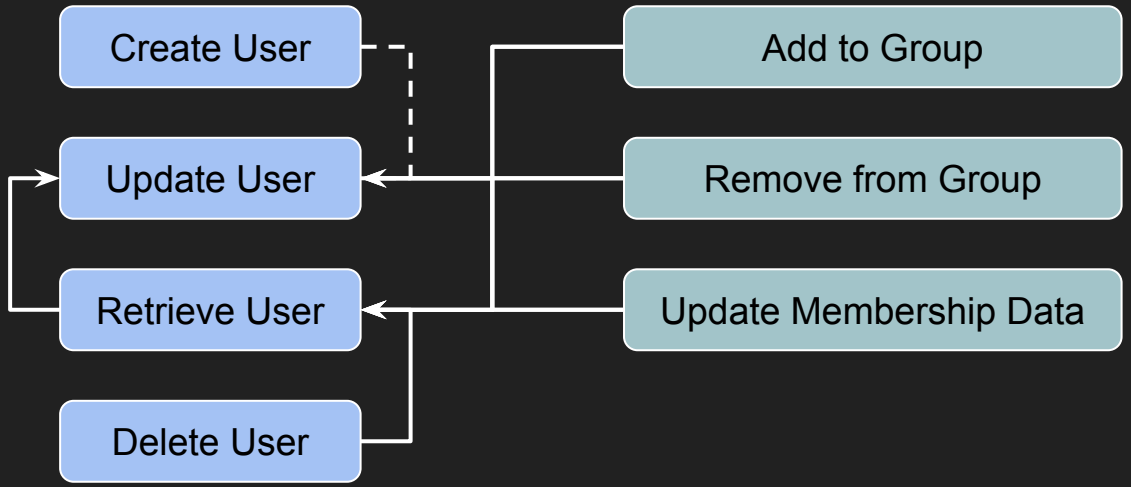
Beware of Declarative Permissions



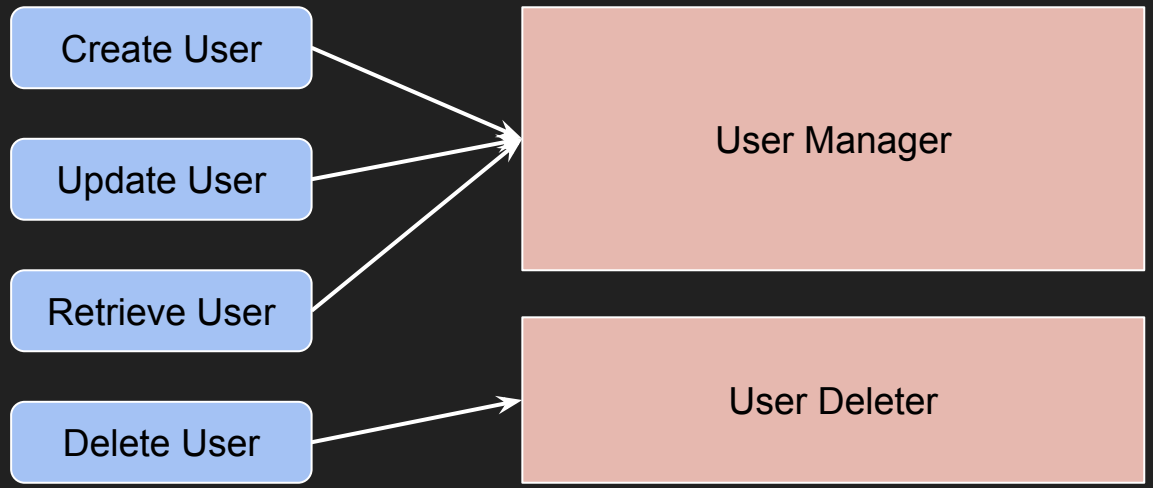
Permission Dependencies



Permission Dependencies



Permission Dependencies



+admin role


```
{{ if (user.hasRole('admin')) }}
  <button>
{{ endif }}
```

Policy Based Access Control

- Also known as PBAC or ABAC (Attribute)
- IF-ELSEIF-ELSE statements

Policy “button-72”

=

‘manager’ && ‘red hair’


```
{{ if (user.inPolicy('button-72')) }}
  <button>
{{ endif }}
```

DaVita has over 300,000 policies!

Auditing

Audit Requirements

- Each standard has different auditing requirements
- Audit logs must be secured (they might contain PII or passwords)
- Audit everything

10/10/2017 12:45:23pm MST

“User admin@inversoft.com added the role ‘user’ to the Application FooBar”

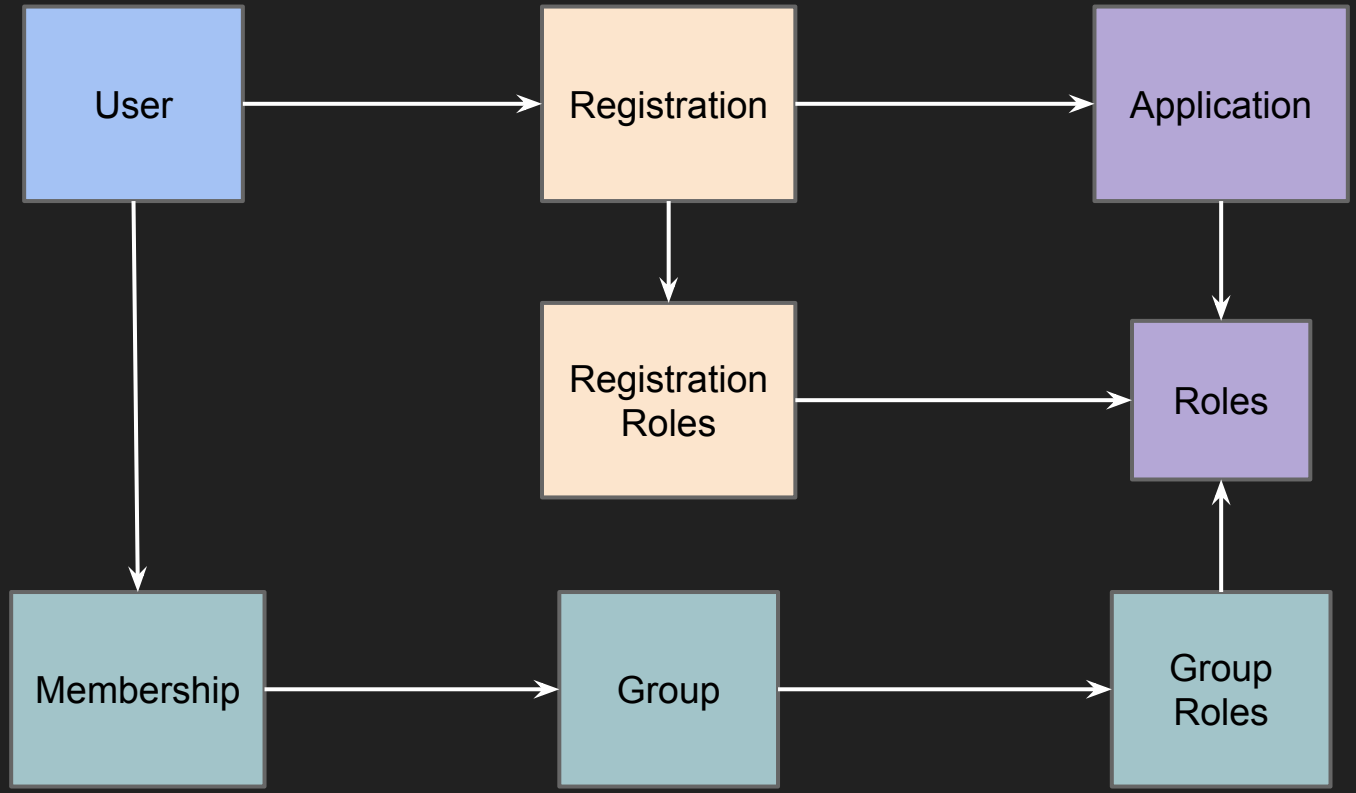
Before:

```
{
  "application": {
    "name": "FooBar",
    "roles": ["admin"]
  }
}
```

After:

```
{
  "application": {
    "name": "FooBar",
    "roles": ["user", "admin"]
  }
}
```

Our Data Model


User
Registration
Application
Registration Roles
Roles
Membership
Group
Group Roles