JWTs:
What developers need to know

Dan Moore

Beer City Code

Aug 2022

@mooreds

THANKS TO ALL OUR SPONSORS!

BizStream

Atomic Object

THE C2 GROUP

KL&A
KUNZ, LEIGH AND ASSOCIATES

MICHIGAN SOFTWARE LABS

WEST MICHIGAN TECH TALENT

TEKTON

KFORCE

LeadingEDJE

Mutually human

Scout APM

Sweetwater
Music Instruments & Pro Audio

About FusionAuth

- FusionAuth is the authentication and authorization platform built for developers, by developers.
- FusionAuth solves the problem of building essential user security without adding risk or distracting from the primary application.

About Me

About Me

- Who cares

What I'll Cover

- JWTs in brief

What I'll Cover

- JWTs in brief
- The problem they solve

What I'll Cover

- JWTs in brief
- The problem they solve
- Validation

What I'll Cover

- JWTs in brief
- The problem they solve
- Validation
- Bearer tokens

What I'll Cover

- JWTs in brief
- The problem they solve
- Validation
- Bearer tokens
- Footguns/common issues

What I'll Cover

- JWTs in brief
- The problem they solve
- Validation
- Bearer tokens
- Footguns/common issues
- Refresh tokens

What I'll Cover

- JWTs in brief
- The problem they solve
- Validation
- Bearer tokens
- Footguns/common issues
- Refresh tokens
- JWT revocation

JWTs
... Briefly

JWTs Briefly

- JSON Web Token

JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’

JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519

JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519
- Signed or encrypted

JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519
- Signed or encrypted
- Stateless, portable tokens of identity

JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519
- Signed or encrypted
- Stateless, portable tokens of identity
- Great for APIs and microservices

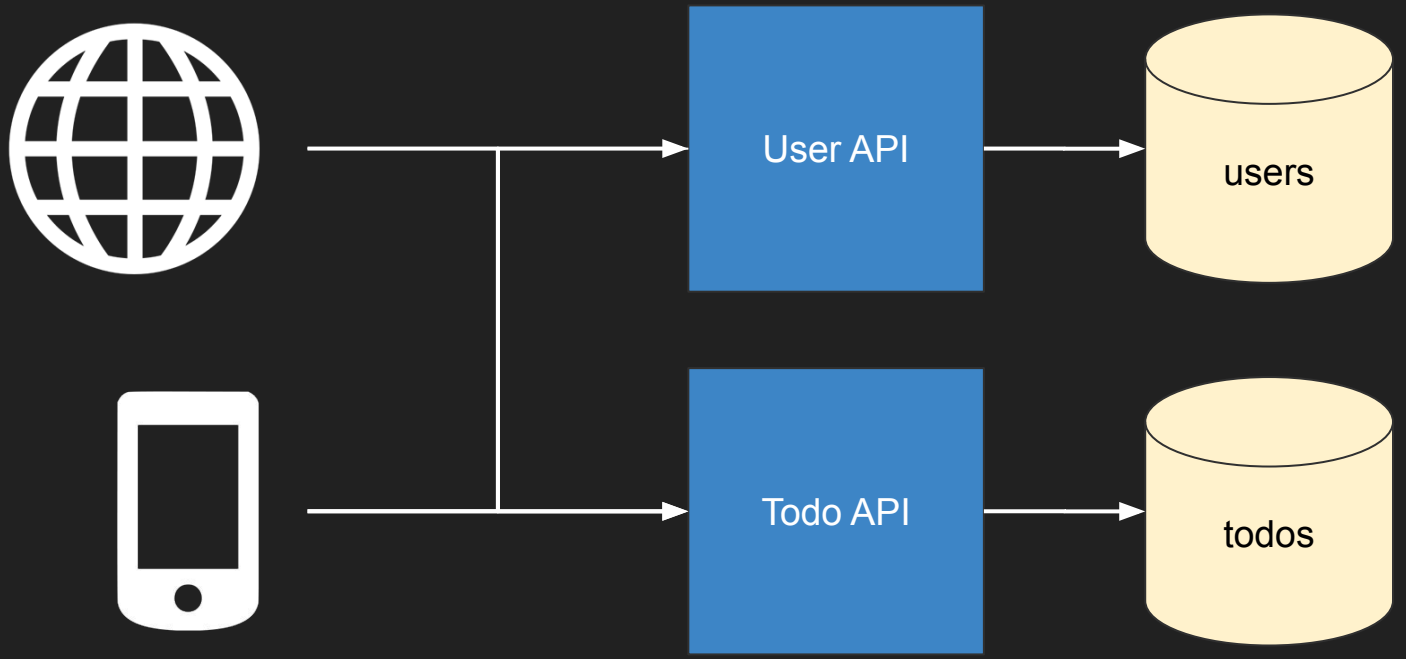
JWTs Briefly

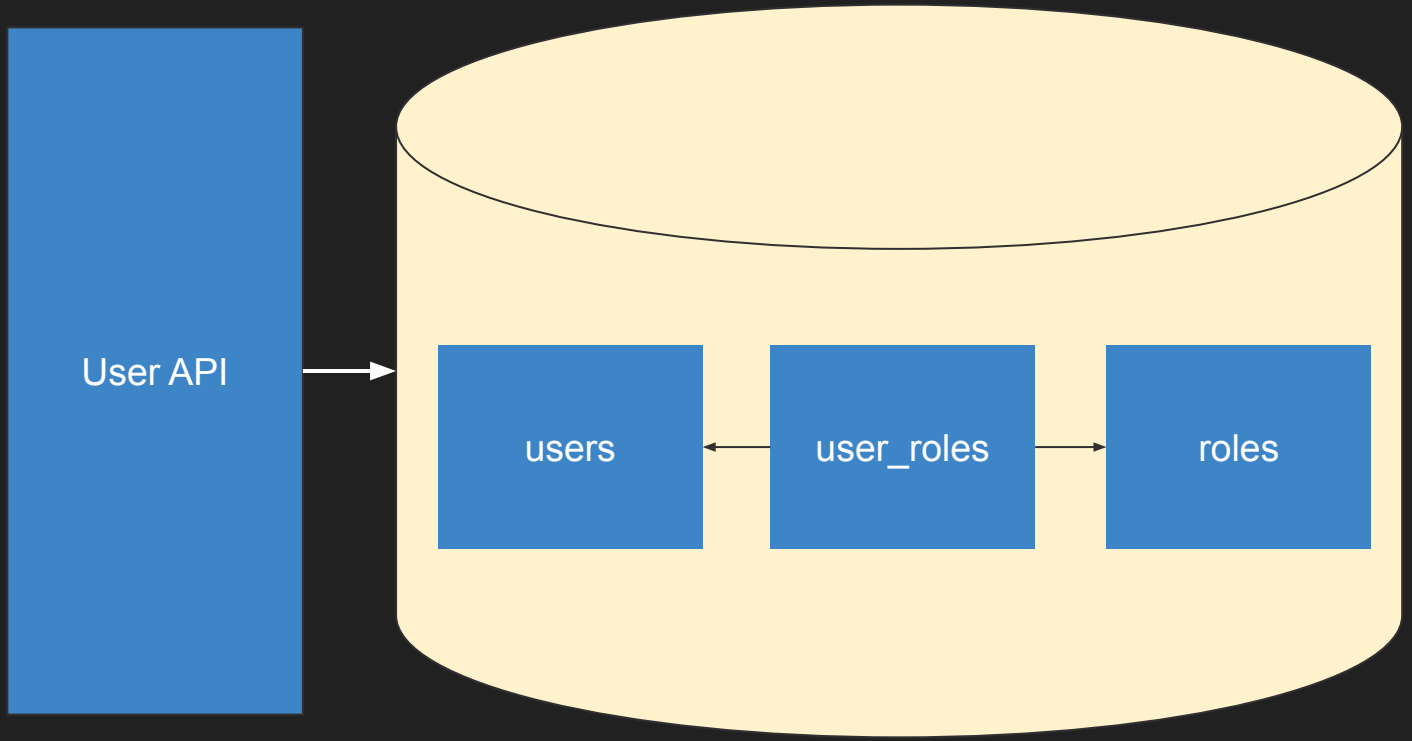
- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519
- Signed or encrypted
- Stateless, portable tokens of identity
- Great for APIs and microservices
- Produced by many Identity Providers

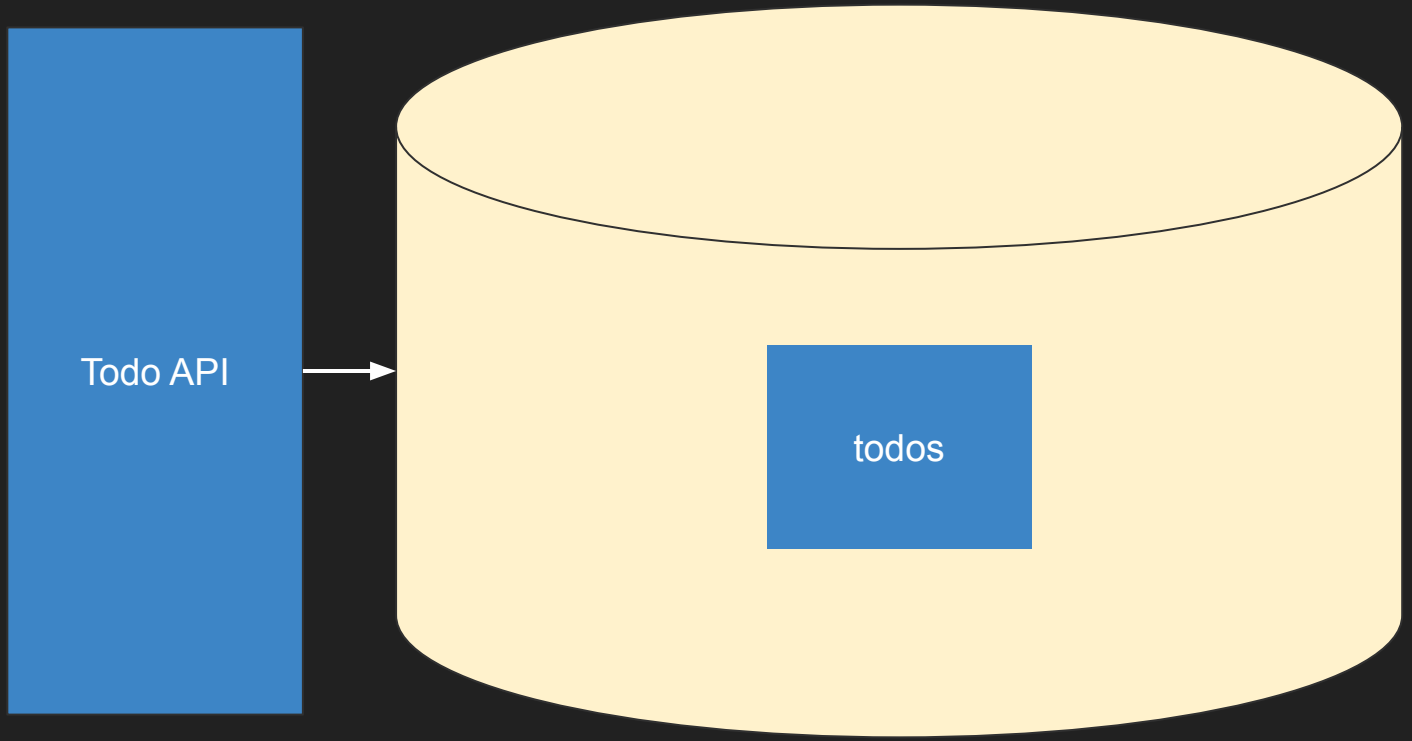
JWTs Briefly

- JSON Web Token
- Pronounced ‘jot’
- Standard - RFC 7519
- Signed or encrypted
- Stateless, portable tokens of identity
- Great for APIs and microservices
- Produced by many Identity Providers
- Widely supported

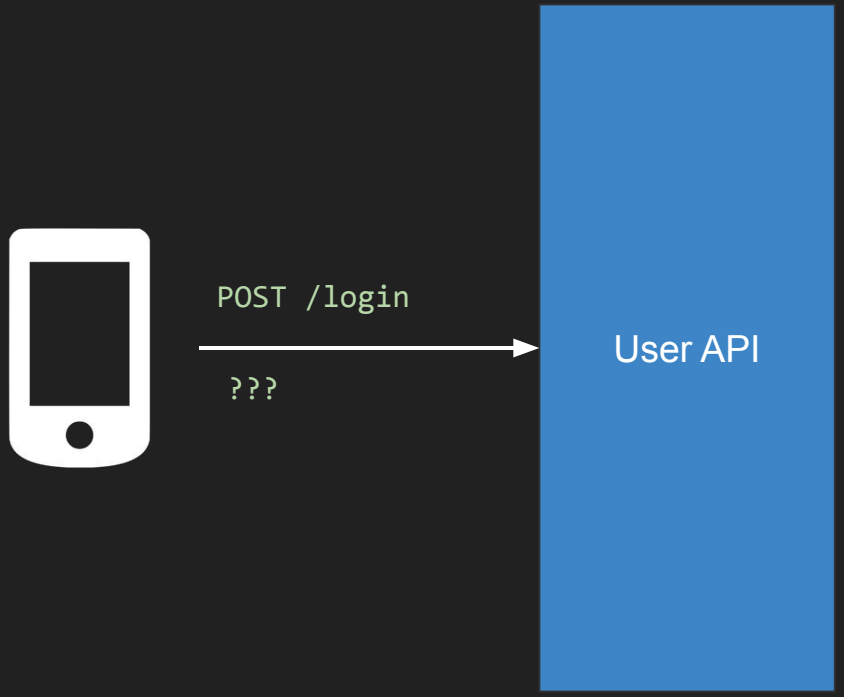
The Problem


User API
users
Todo API
todos

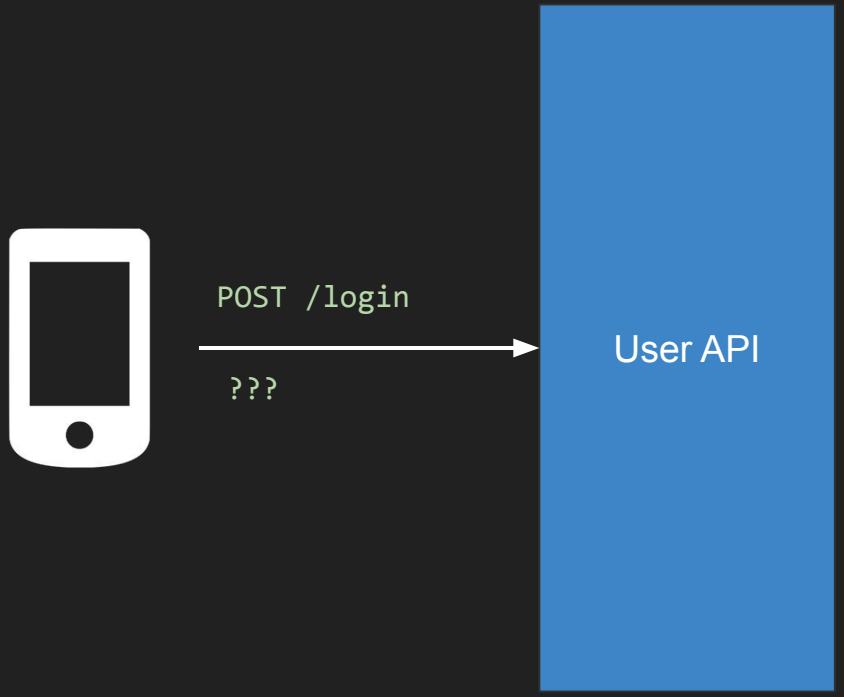

User API
users
user_roles
roles

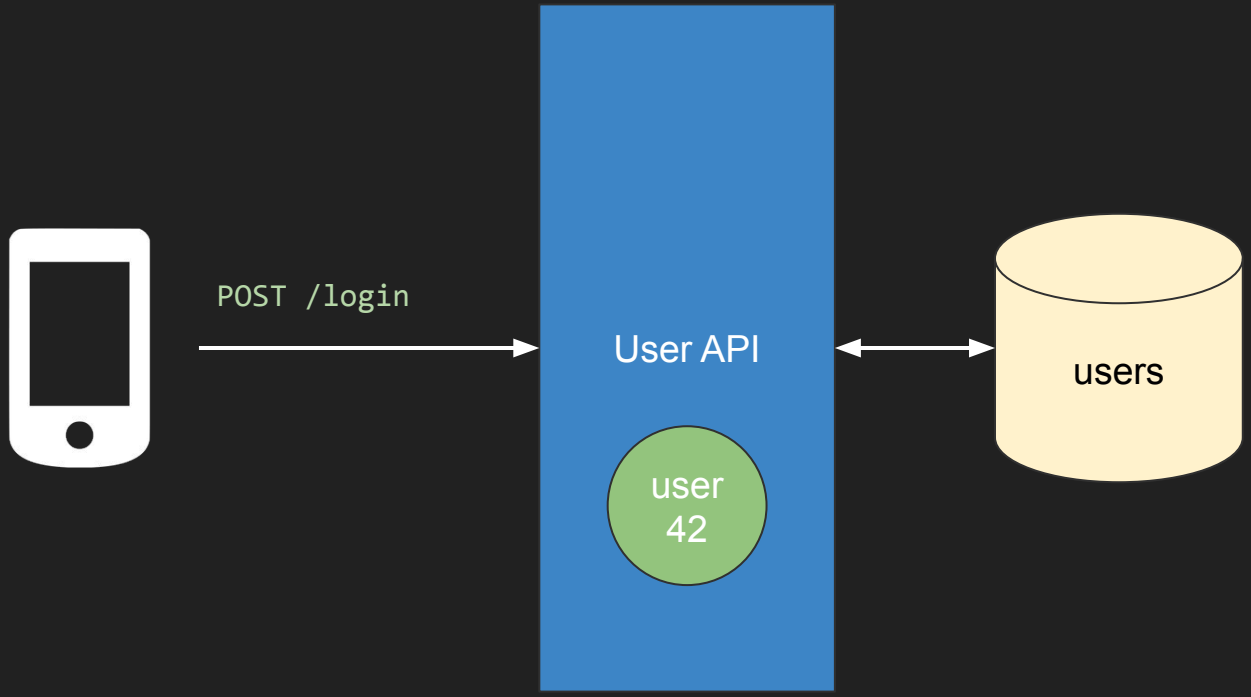

Todo API
todos

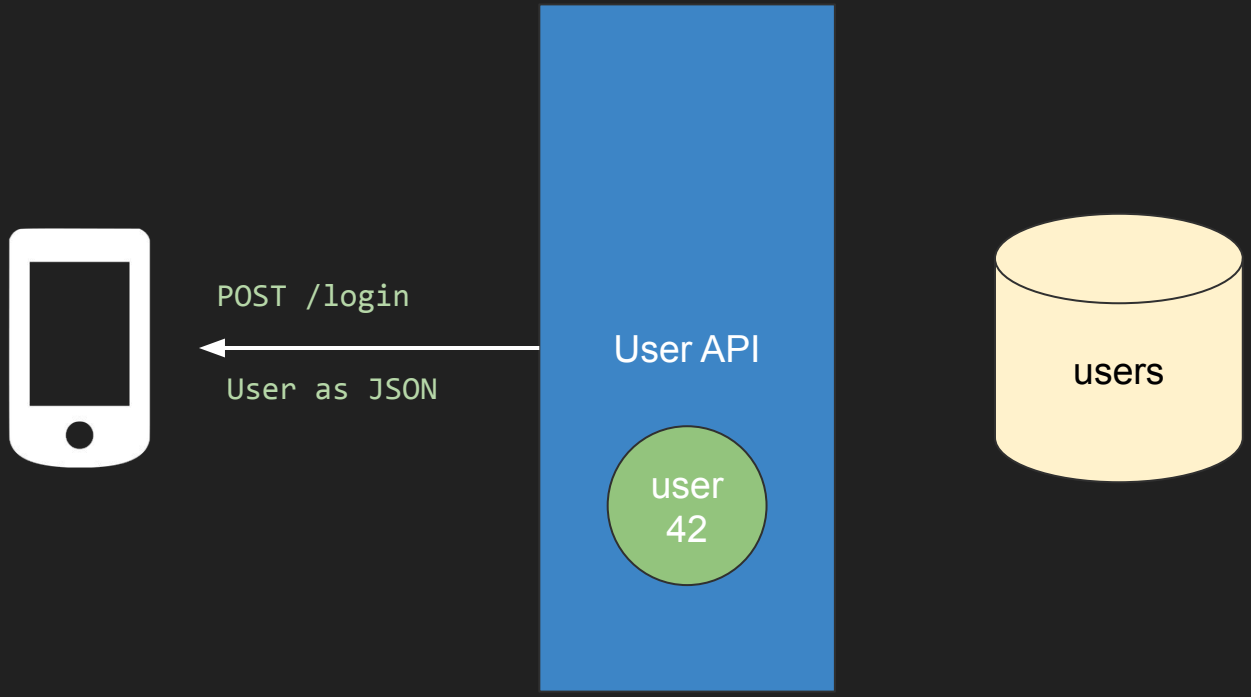
```
CREATE TABLE todos (
  id INT NOT NULL,
  text TEXT NOT NULL,
  status INT NOT NULL,
  user_id CHAR(40) NOT NULL,
  PRIMARY KEY (id)
);
```


POST /login
???
User API

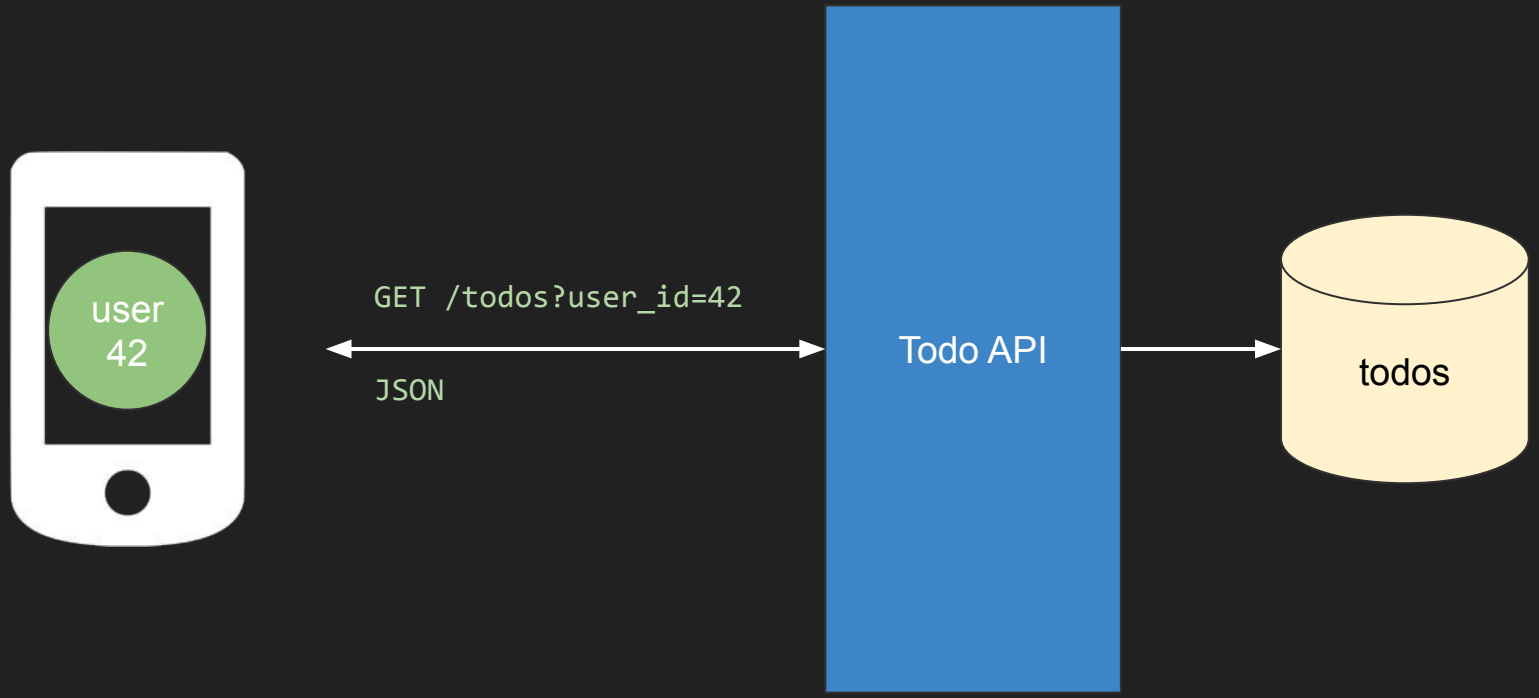
Solutions


POST /login
???
User API


POST /login
User API
user
42
users

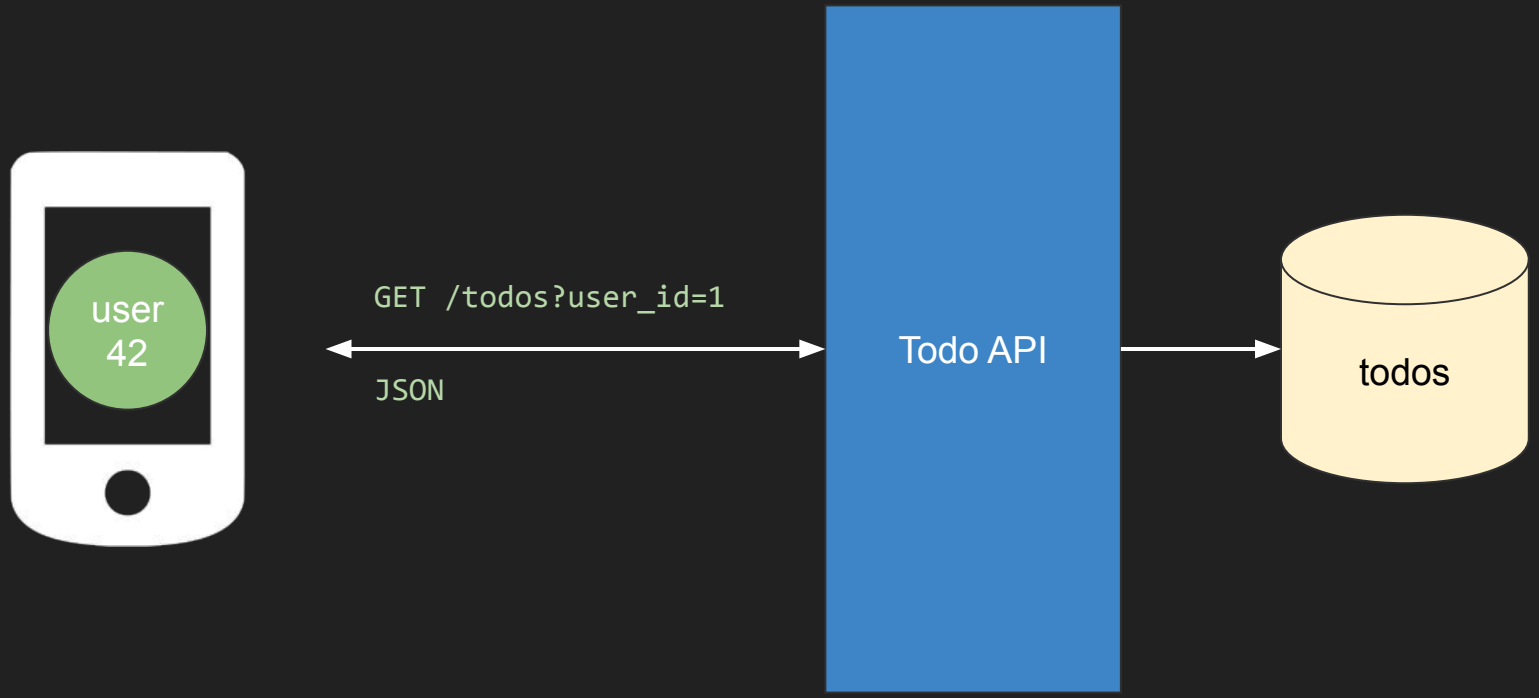

POST /login
User as JSON
User API
user
42
users

```
{
  "user": {
    "id": "42",
    "name": "Dan Moore",
    "email": "dan@fusionauth.io",
    "roles": ["admin"]
  }
}
```

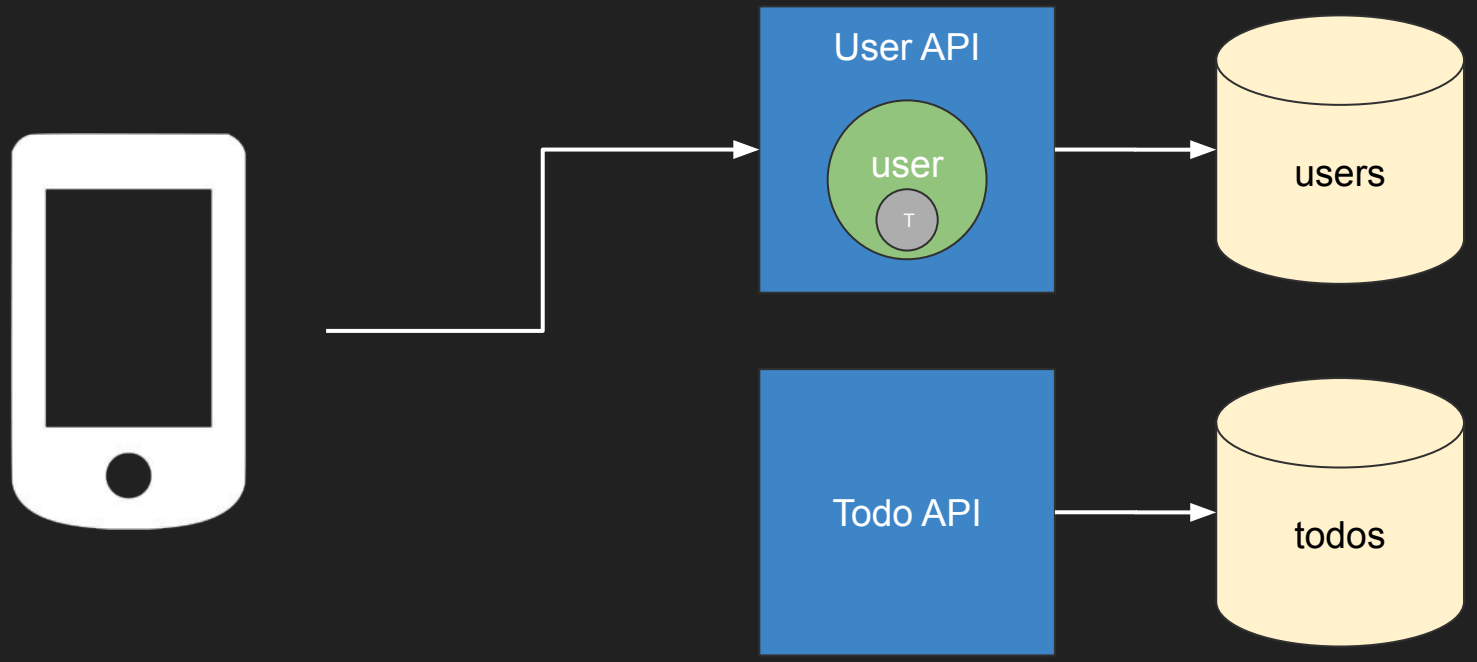

user
42
GET /todos?user_id=42
JSON
Todo API
todos

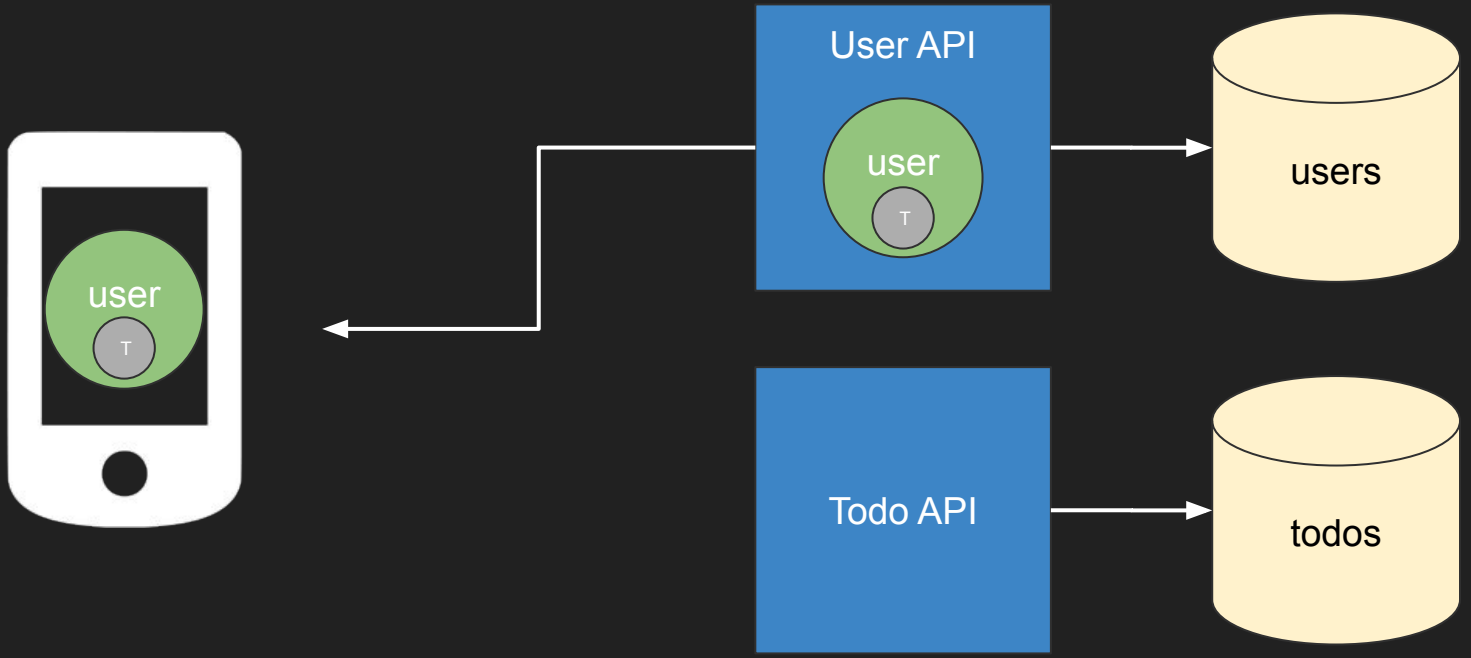
Good
Idea?

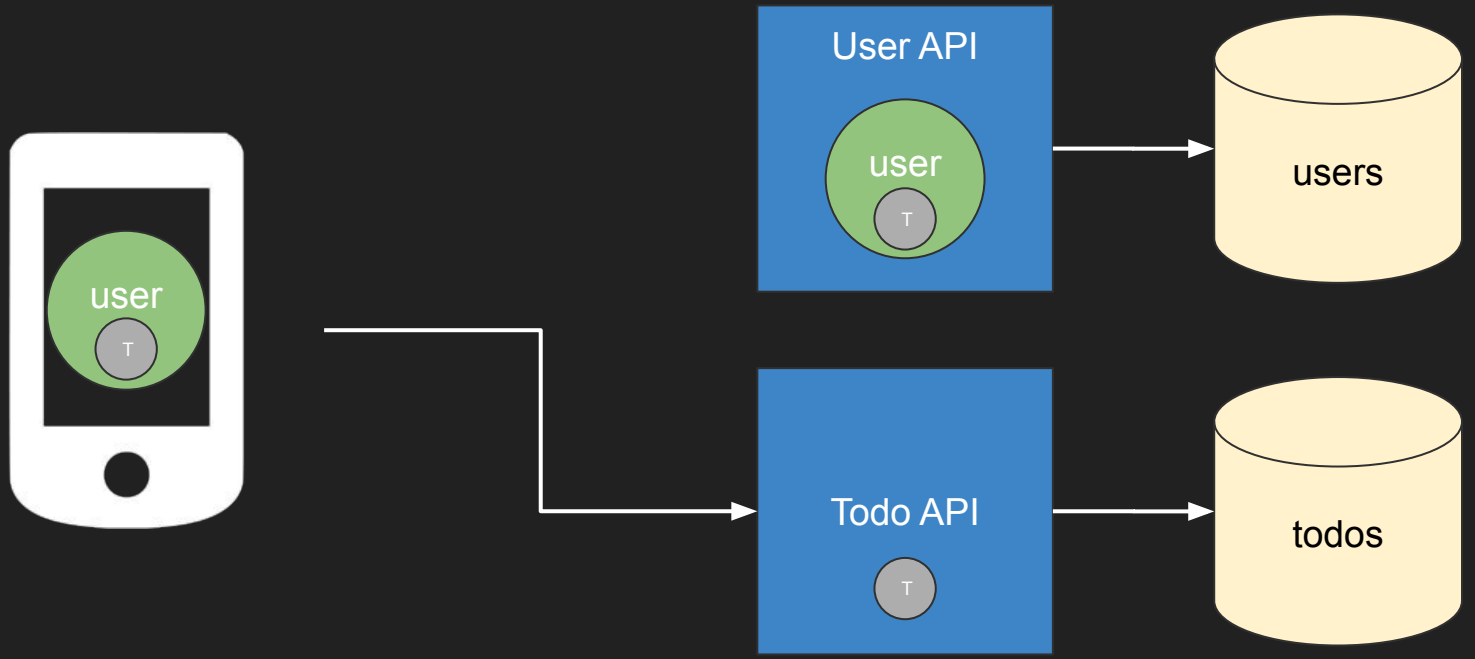
Danger Will Robinson!!

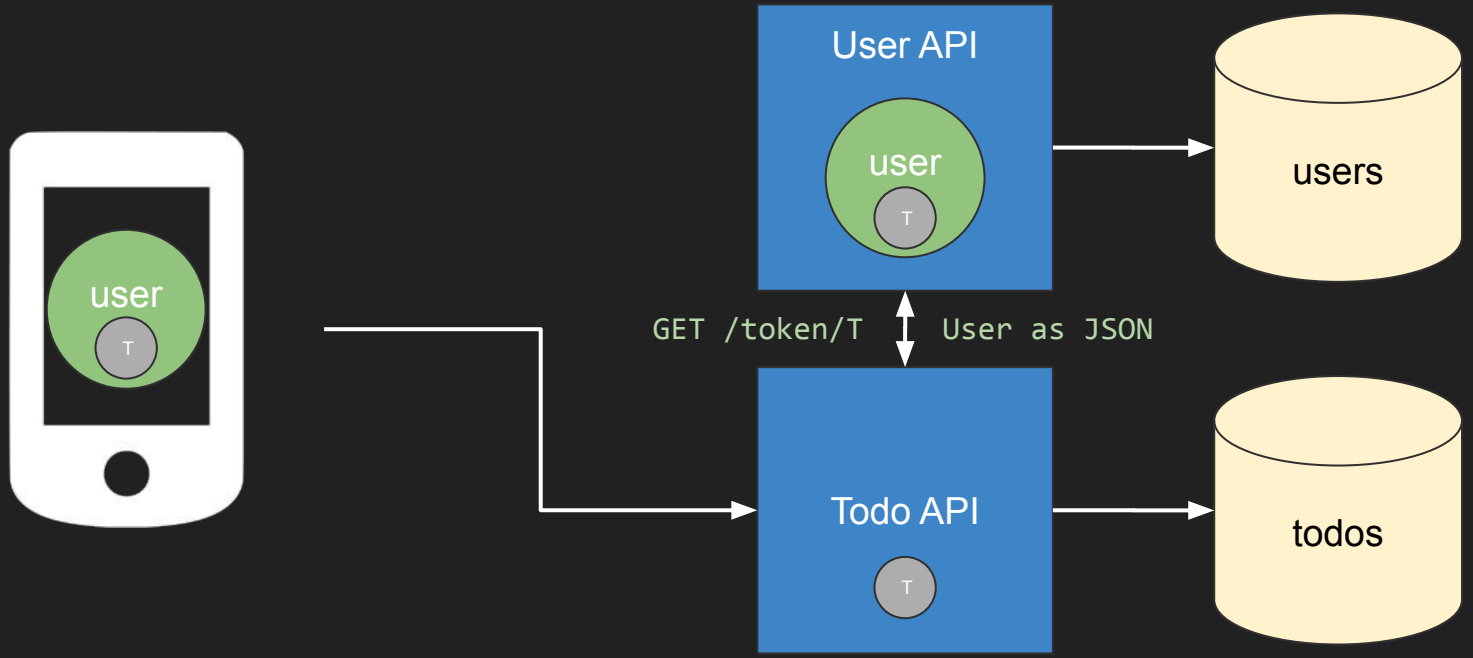

user
42
GET /todos?user_id=1
JSON
Todo API
todos

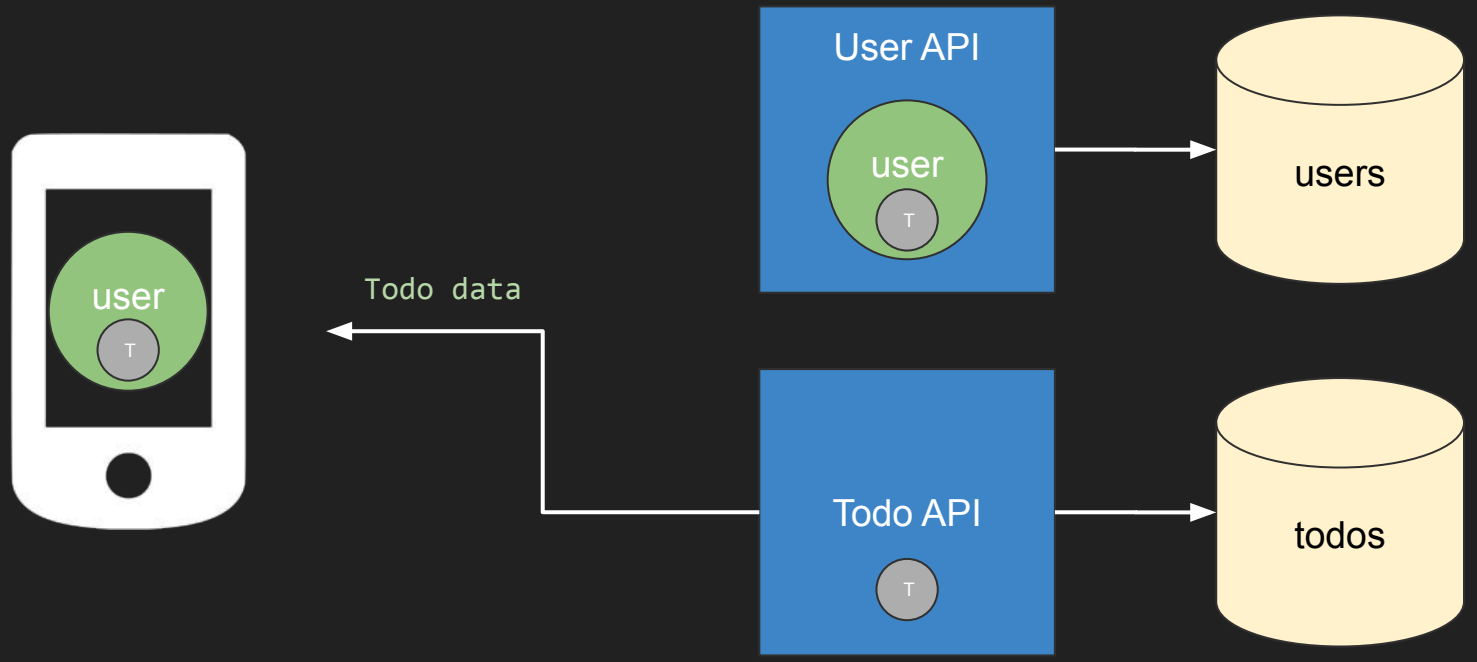
A Different
Approach


User API
user
T
users
Todo API
todos


User API
user
T
users
user
T
Todo API
todos


User API
user
T
users
user
T
Todo API
T
todos

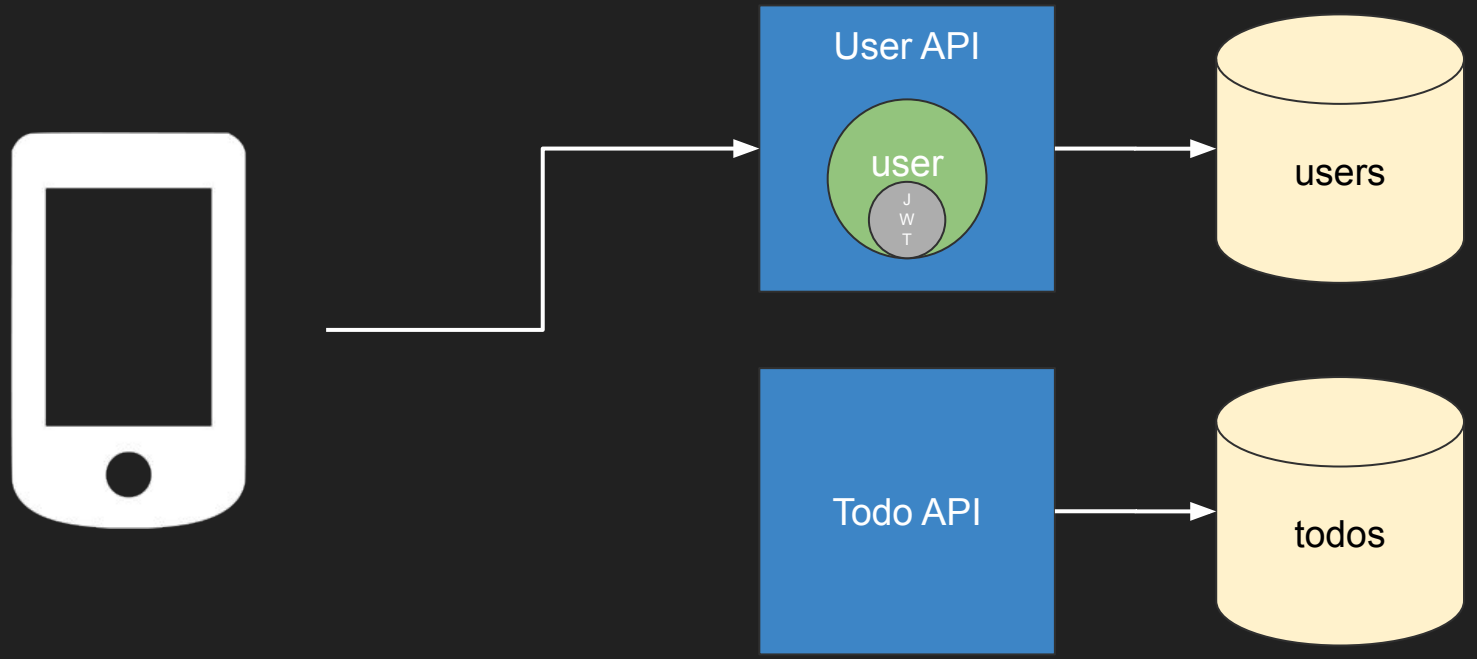

User API
user
T
users
user
T
GET /token/T
User as JSON
Todo API
T
todos

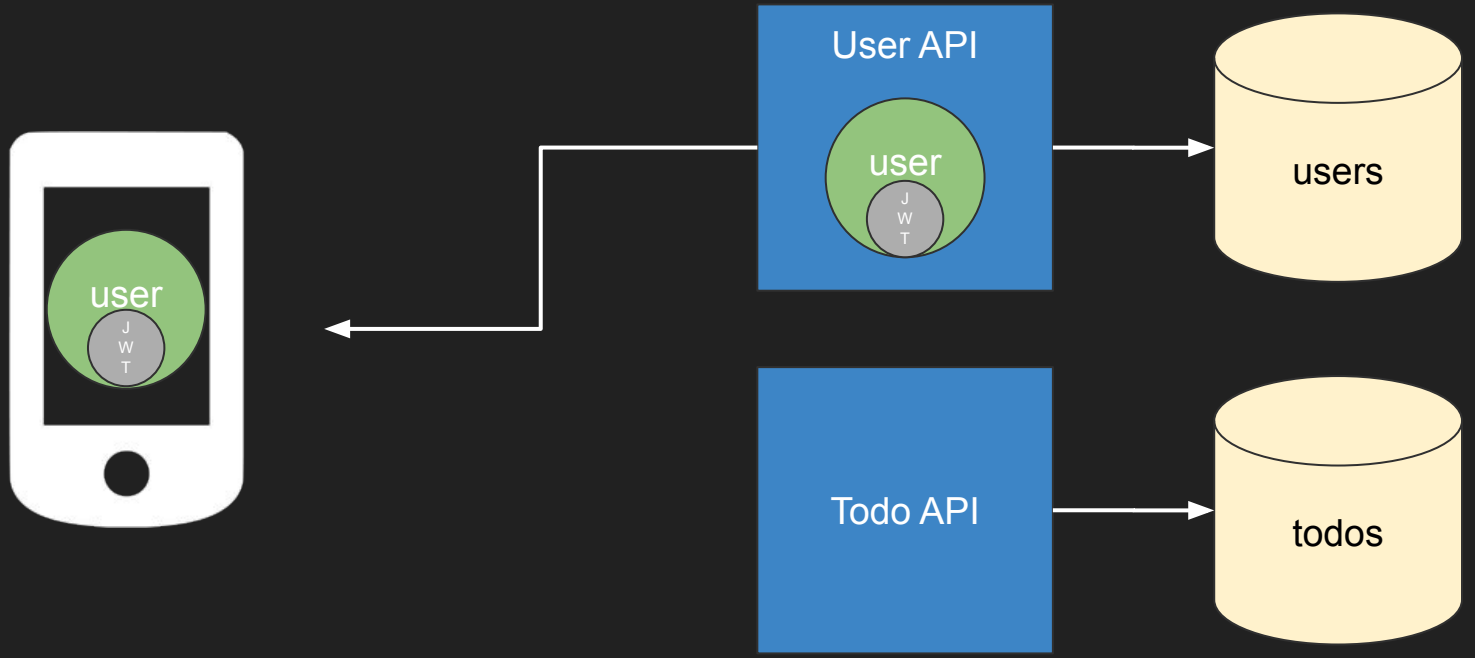

User API
user
T
users
user
T
Todo data
Todo API
T
todos

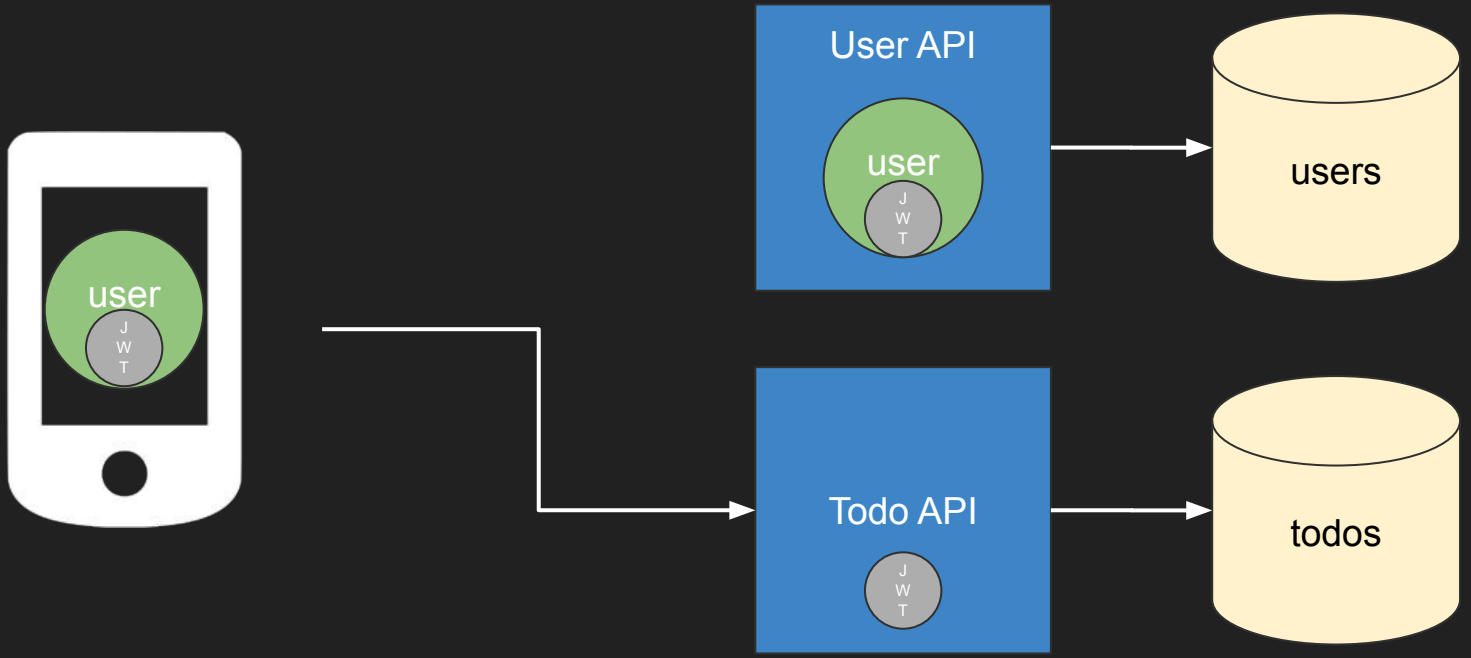
Opaque Tokens

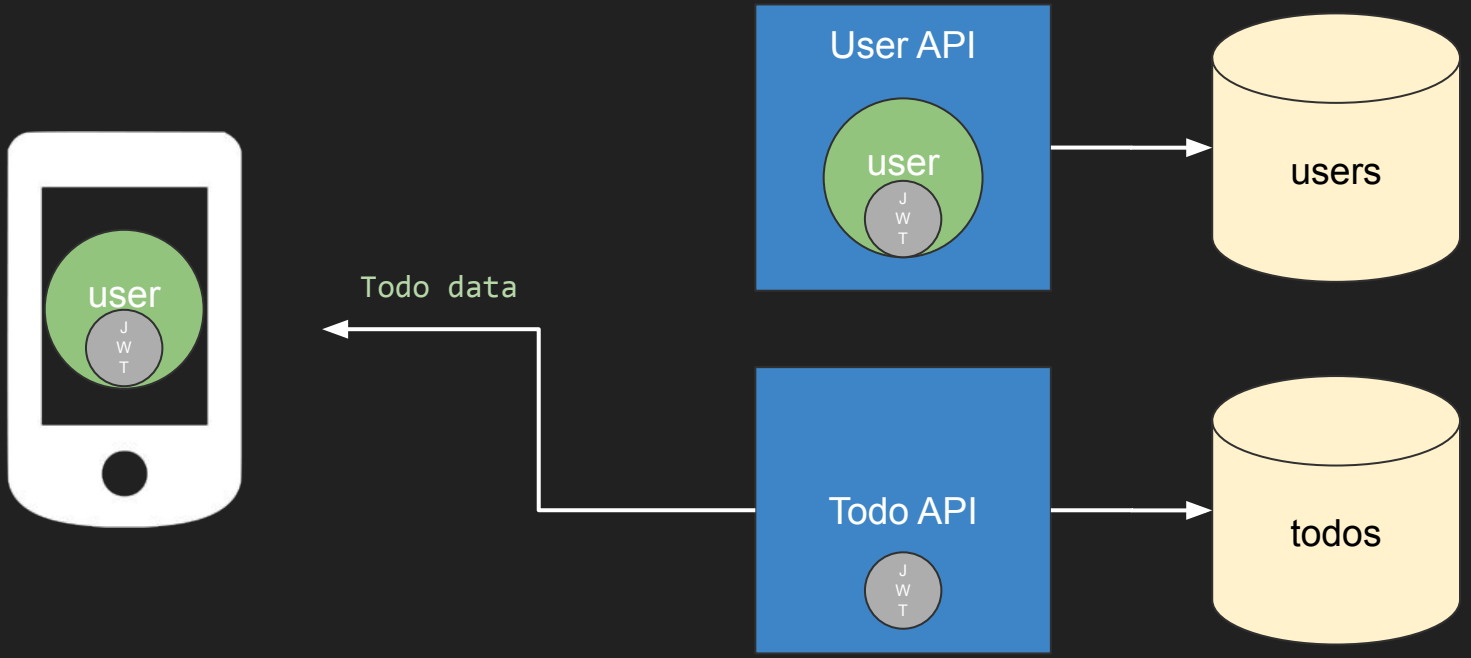
- There must be a User -> Token mapping
- In memory or in database
- Can be very chatty
- Couples the User API to EVERYTHING (almost)
- This is OAuth Introspection (RFC 7662)

JWTs To
The Rescue


User API
user
J
W
T
users
Todo API
todos


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


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


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

JWT Benefits

- Signed
  - Public/private key pair
  - Symmetric key

JWT Benefits

- Signed
  - Public/private key pair
  - Symmetric key
- Validated by Todo API without calling User API

JWT Benefits

- Signed
  - Public/private key pair
  - Symmetric key
- Validated by Todo API without calling User API
- Contains roles and other metadata

JWTs
Under the Microscope

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJmdXNpb25hdXRoLml
vIiwiZXhwIjoxNjE5NTU1MDE4LCJhdWQiOiIyMzhkNDc5My03MGRlLTQxODMtO
TcwNy00OGVkOGVjZDE5ZDkiLCJzdWIiOiIxOTAxNmI3My0zZmZhLTRiMjYtODB
kOC1hYTkyODc3Mzg2NzciLCJuYW1lIjoiRGFuIE1vb3JlIiwibWVtYmVyc2hpc
EV4cGlyZWQiOmZhbHNlLCJyb2xlcyI6WyJSRVRSSUVWRV9UT0RPUyIsIkFETUl
OIl19.cPL36Al_8eT7YQVowIOruitxXb0n8w4DKaWVthfEwfc

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9

=

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

eyJpc3MiOiJmdXNpb25hdXRoLmlvIiwiZXhwIjoxNjE5NTU1MDE4LCJhdWQiOiIyMzhkNDc5My03MGRlLTQxODMtOTcwNy00OGVkOGVjZDE5ZDkiLCJzdWIiOiIxOTAxNmI3My0zZmZhLTRiMjYtODBkOC1hYTkyODc3Mzg2NzciLCJuYW1lIjoiRGFuIE1vb3JlIiwibWVtYmVyc2hpcEV4cGlyZWQiOmZhbHNlLCJyb2xlcyI6WyJSRVRSSUVWRV9UT0RPUyIsIkFETUlOIl19

=

```
{
  "iss": "fusionauth.io",
  "exp": 1619555018,
  "aud": "238d4793-70de-4183-9707-48ed8ecd19d9",
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Dan Moore",
  "roles": ["RETRIEVE_TODOS", "ADMIN"]
}
```

cPL36Al_8eT7YQVowIOruitxXb0n8w4DKaWVthfEwfc

=

Signature

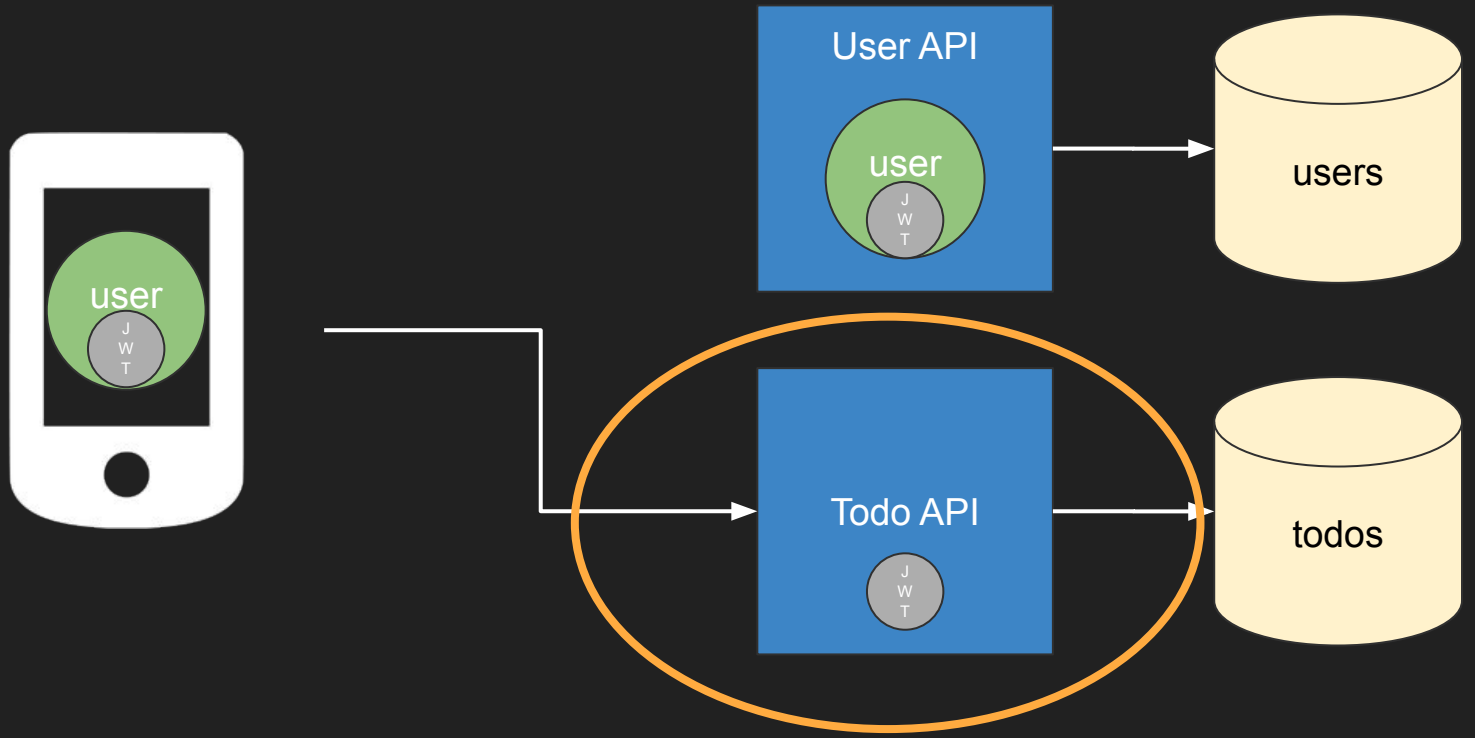
Signature Generation

- Cryptographic operation(Header + Body)

Signature Generation

- Cryptographic operation(Header + Body)
- Operation varies by algorithm

JWT
Validation


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

Validate The Signature

Validate The Signature

- Use a library

Validate The Signature

- Use a library
- If it doesn't match, **STOP**

Validate The Claims

Validate The Claims

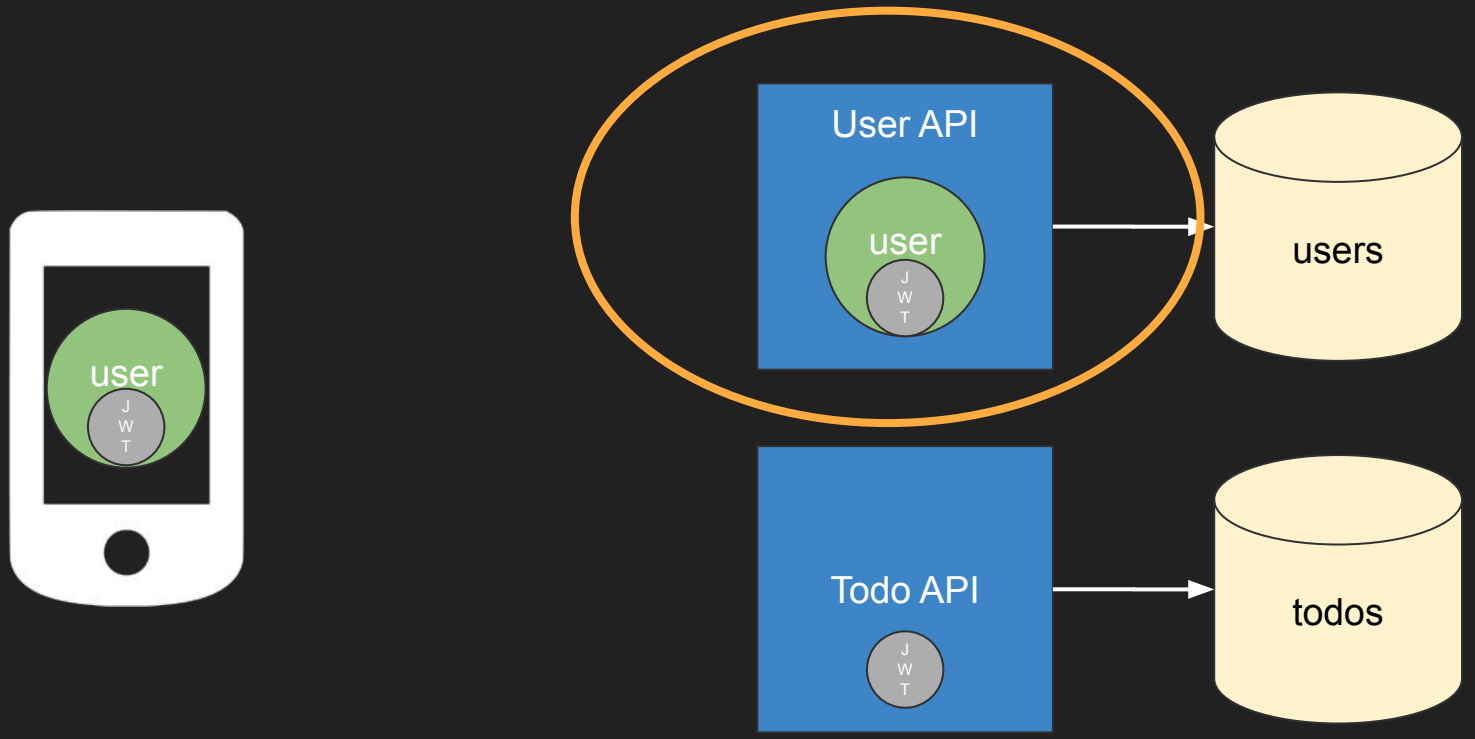
- Business logic

Validate The Claims

- Business logic
- **You must write**

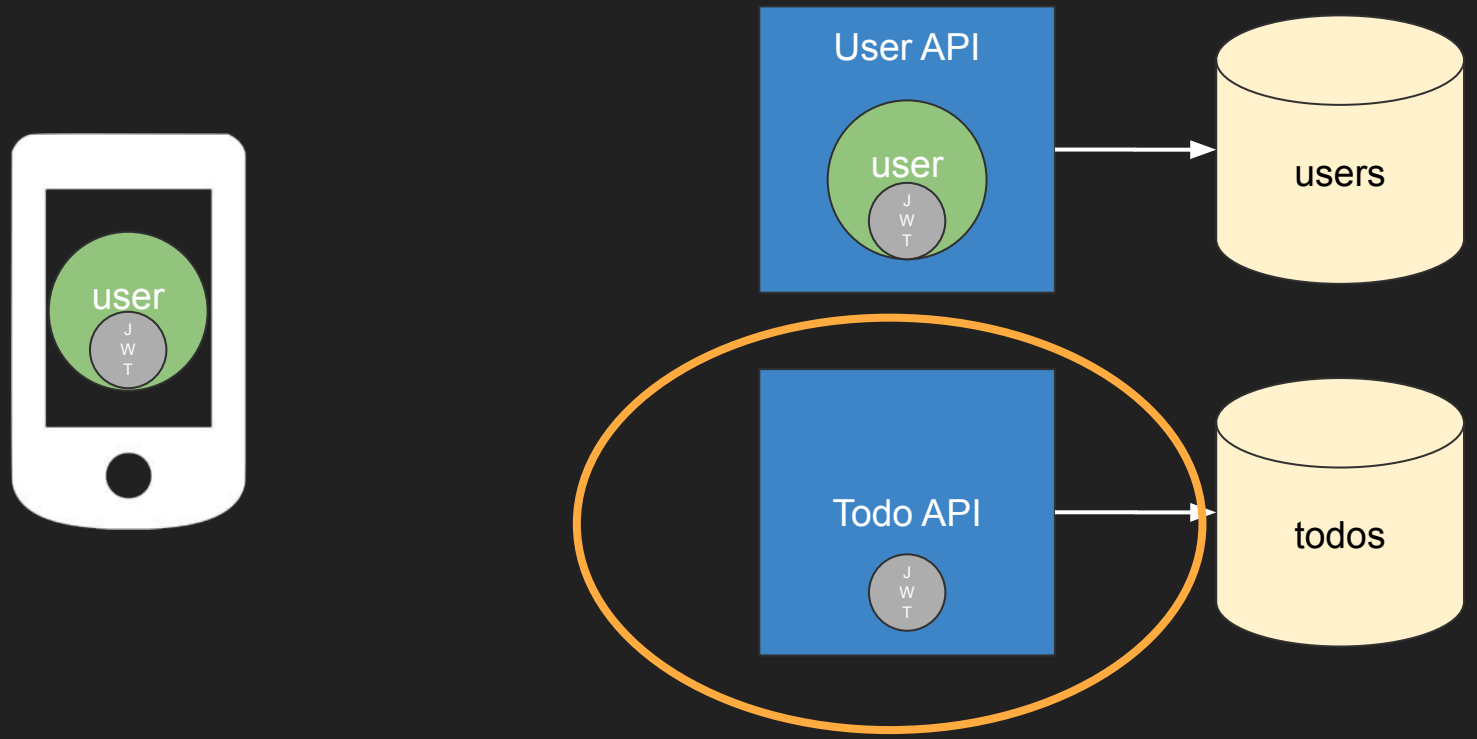
```
{
  "iss": "fusionauth.io",
  "exp": 1619555018,
  "aud": "238d4793-70de-4183-9707-48ed8ecd19d9",
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Dan Moore",
  "roles": ["RETRIEVE_TODOS", "ADMIN"]
}
```

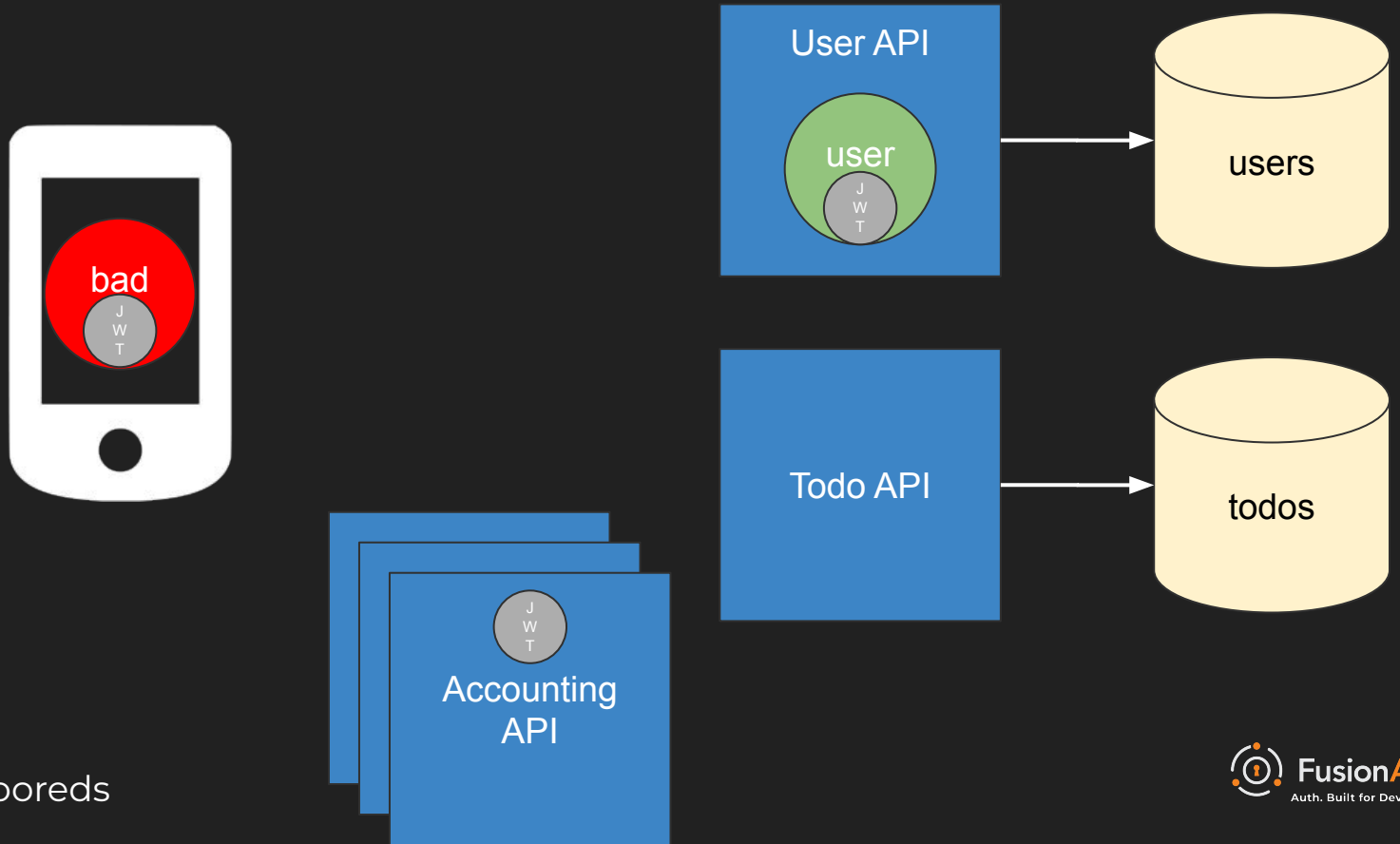
```
{
  "iss": "fusionauth.io",
  "exp": 1619555018,
  "aud": "238d4793-70de-4183-9707-48ed8ecd19d9",
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Dan Moore",
  "roles": ["RETRIEVE_TODOS", "ADMIN"]
}
```

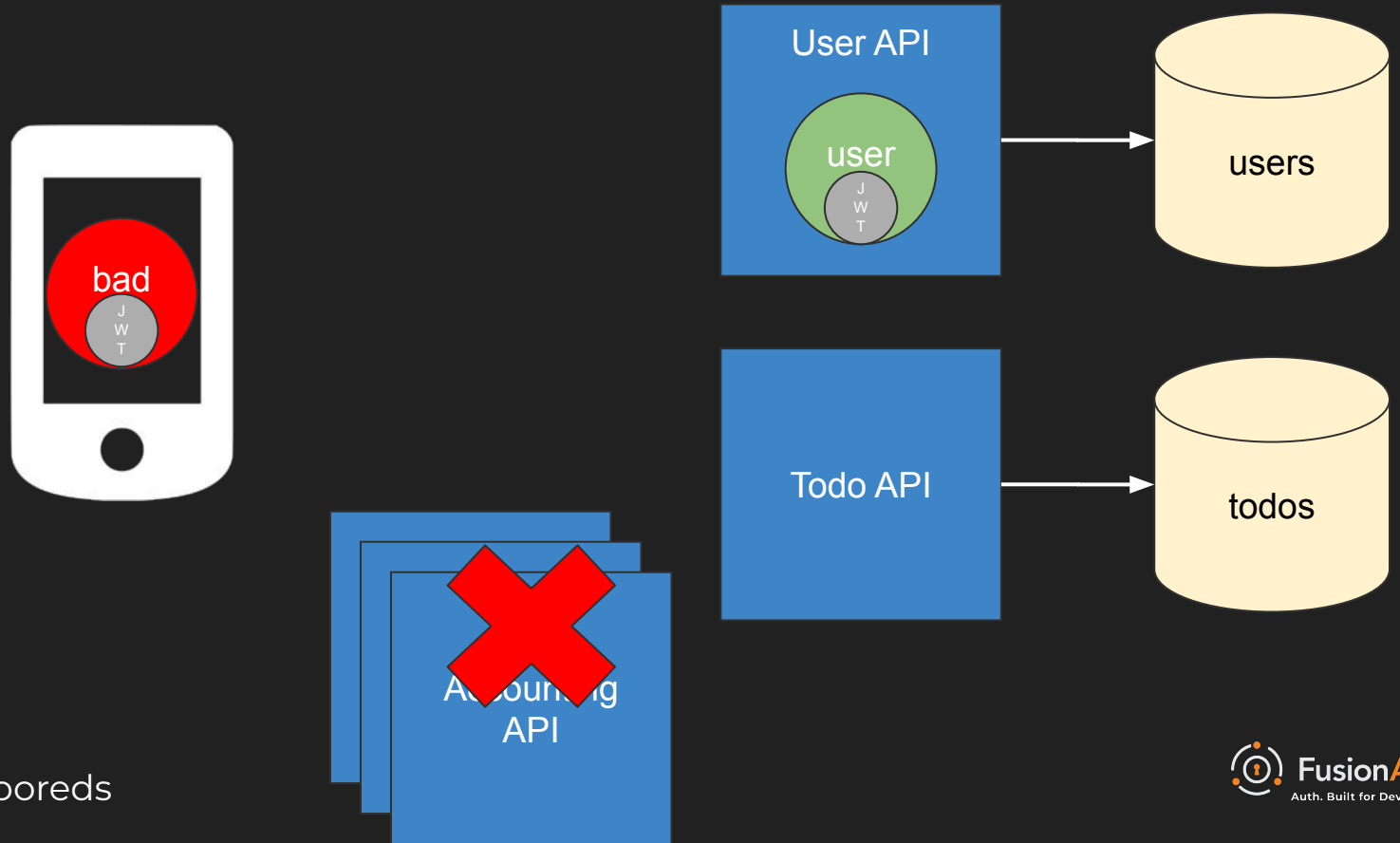

user
J
W
T
User API
user
J
W
T
users
Todo API
J
W
T
todos

```
{
  "iss": "fusionauth.io",
  "exp": 1619555018,
  "aud": "238d4793-70de-4183-9707-48ed8ecd19d9",
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Dan Moore",
  "roles": ["RETRIEVE_TODOS", "ADMIN"]
}
```

```
{
  "iss": "fusionauth.io",
  "exp": 1619555018,
  "aud": "238d4793-70de-4183-9707-48ed8ecd19d9",
  "sub": "19016b73-3ffa-4b26-80d8-aa9287738677",
  "name": "Dan Moore",
  "roles": ["RETRIEVE_TODOS", "ADMIN"]
}
```

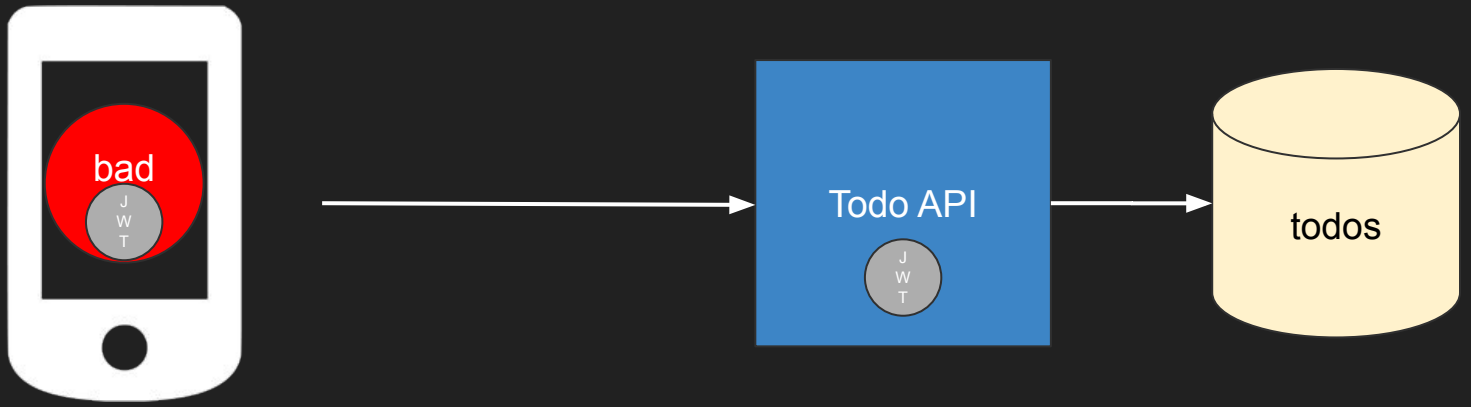

user
J
W
T
User API
user
J
W
T
users
Todo API
J
W
T
todos


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


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

Code

Bearer
Tokens


bad
J
W
T
Todo API
J
W
T
todos

Token Storage

- Prefer server side
  - Known as the BFF, or backend for front end, pattern

Token Storage

- Prefer server side
  - Known as the BFF, or backend for front end, pattern
- HttpOnly, secure cookie in browser
  - Watch out for XSS

Token Storage

- Prefer server side
  - Known as the BFF, or backend for front end, pattern
- HttpOnly, secure cookie in browser
  - Watch out for XSS
- Secure storage on mobile

Common
Issues

Incorrect Validation

- Use a library
  - There are lots out there
- Check your claims
  - Standard
  - Custom

Putting Secrets In a JWT

- Base64 encoded JSON
  - Integrity guaranteed
  - Not secrecy
- Can leak information

Using HMAC Without a Good Key

- Susceptible to brute force
  - https://github.com/brendan-rius/c-jwt-cracker

Using The “none” Algorithm

- No signature

Using The “none” Algorithm

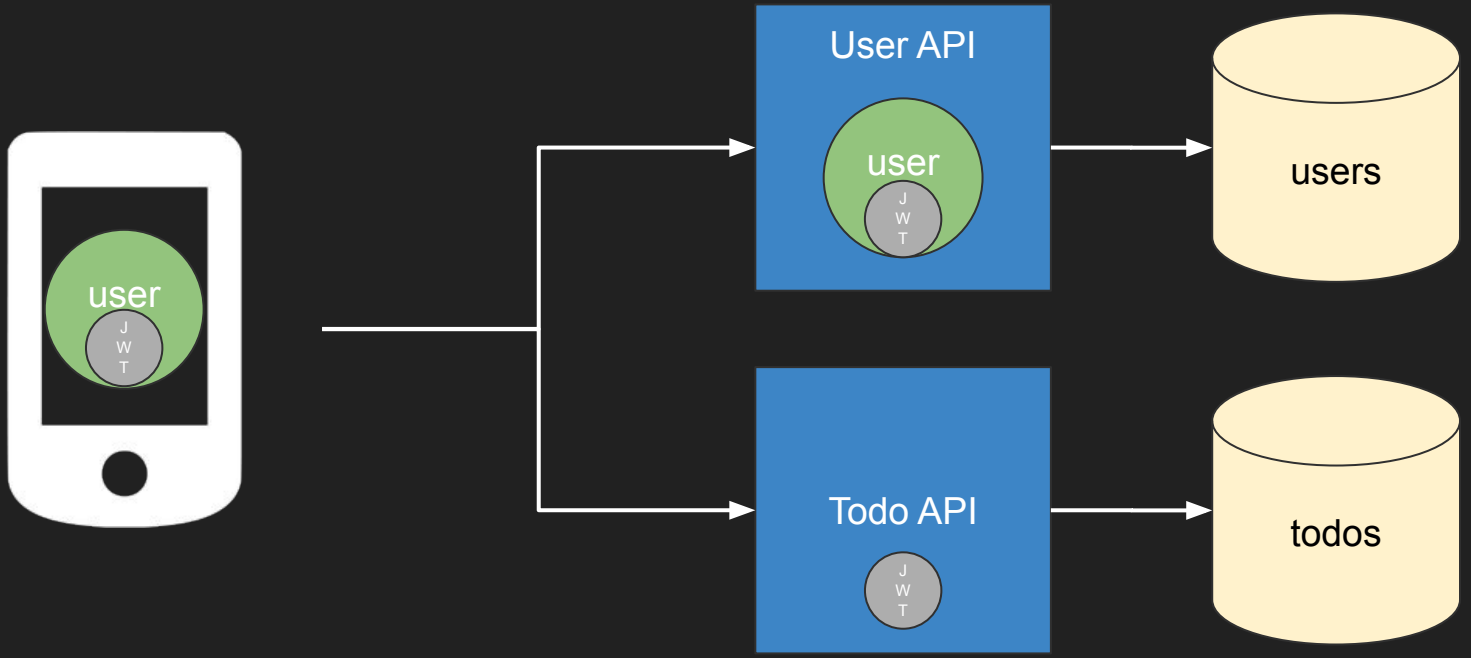
- No signature == no integrity check

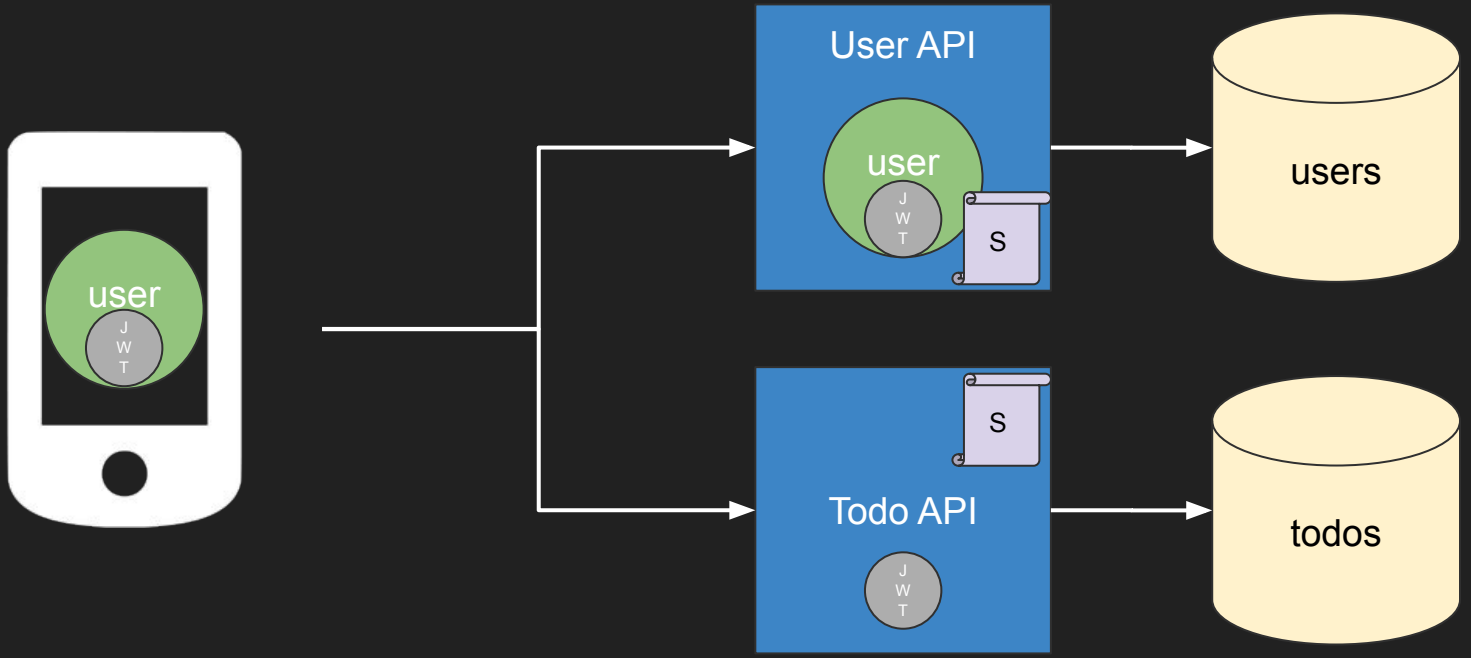
eyJhbGciOiJub25lIn0.eyJpc3MiOiJmdXNpb25hdXRoLmlvIiwiZXhwIjoxNTg5MjI3NDgwLCJhdWQiOiIyMzhkNDc5My03MGRlLTQxODMtOTcwNy00OGVkOGVjZDE5ZDkiLCJzdWIiOiIxOTAxNmI3My0zZmZhLTRiMjYtODBkOC1hYTkyODc3Mzg2NzciLCJuYW1lIjoiRGFuIE1vb3JlIiwicm9sZXMiOlsiUkVUUklFVkVfVE9ET1MiXX0.

Using The “none” Algorithm

- Unsanitized credentials!

Asymmetric
Keypairs


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

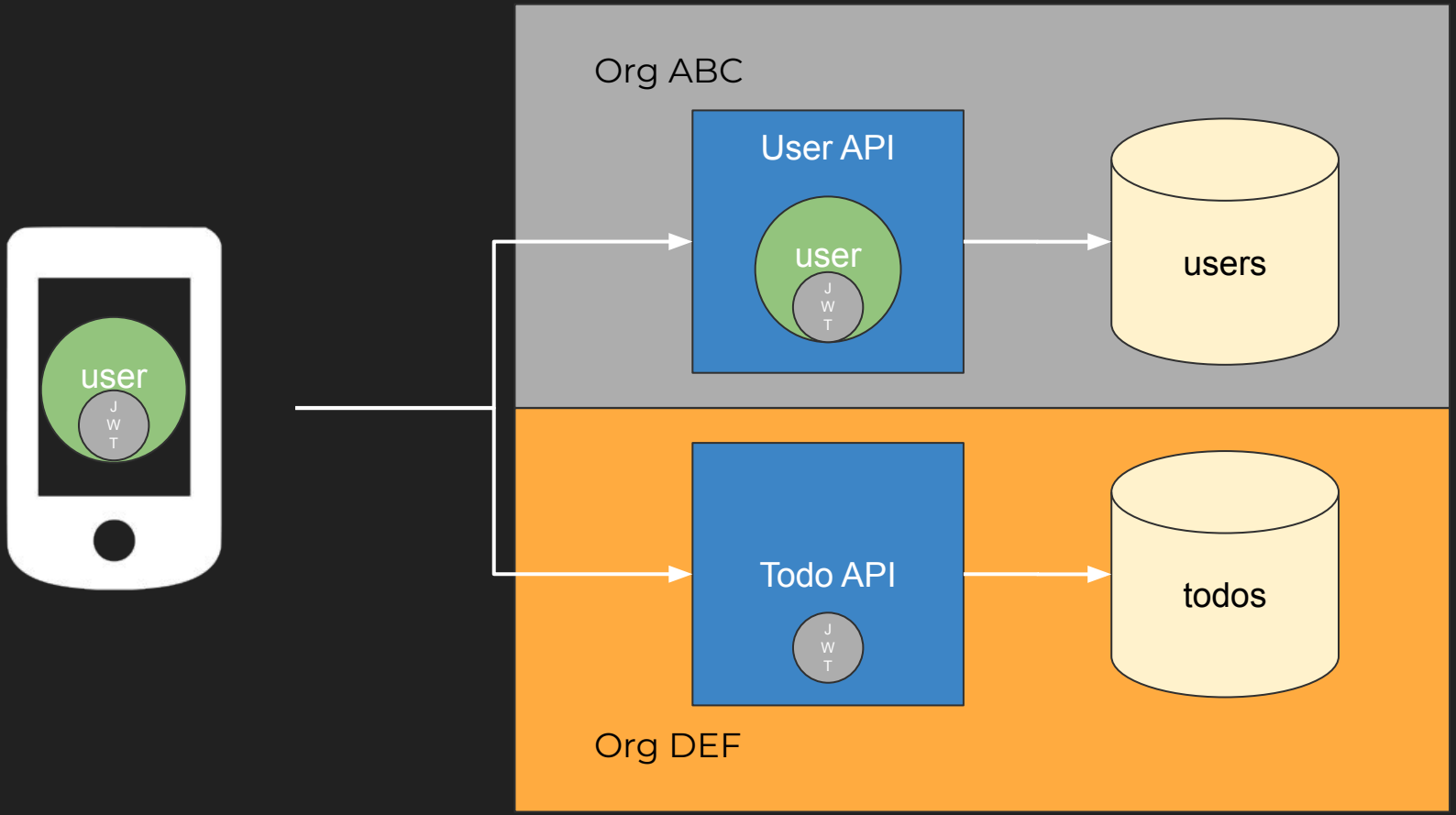

user
J
W
T
User API
user
J
W
T
S
users
S
Todo API
J
W
T
todos

Asymmetric Keypairs Avoid This

- The private key signs the JWT
- The public key is used to verify it

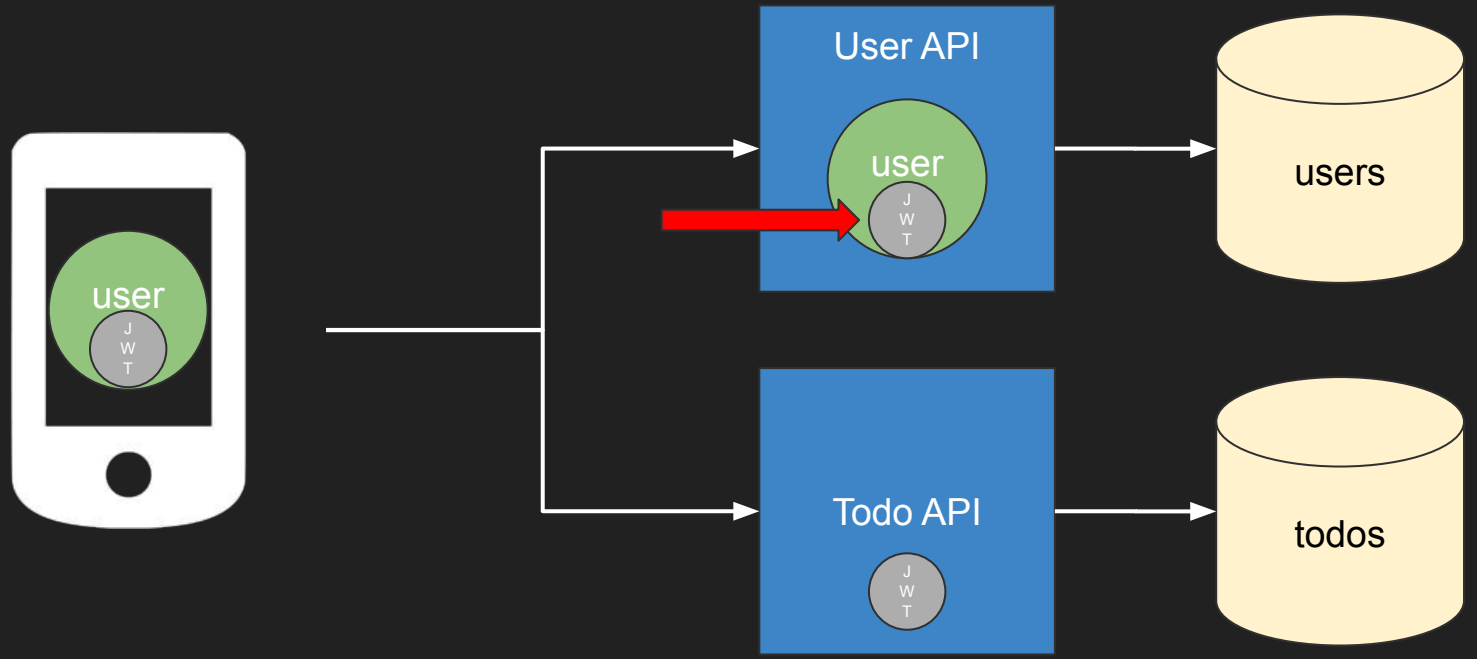
Why does that matter?

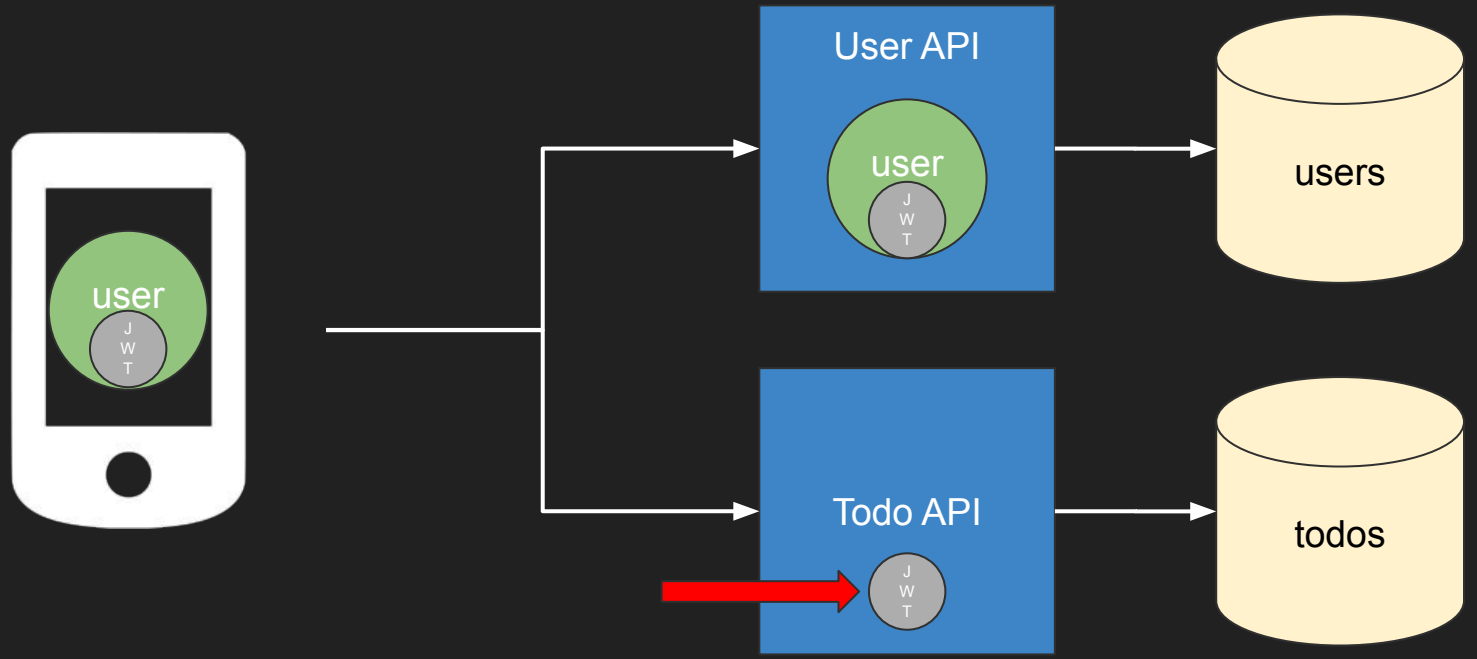
- Scale


Org ABC
User API
user
J
W
T
users
user
J
W
T
Todo API
J
W
T
todos
Org DEF

Why does that matter?

- Scale
- Security


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


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

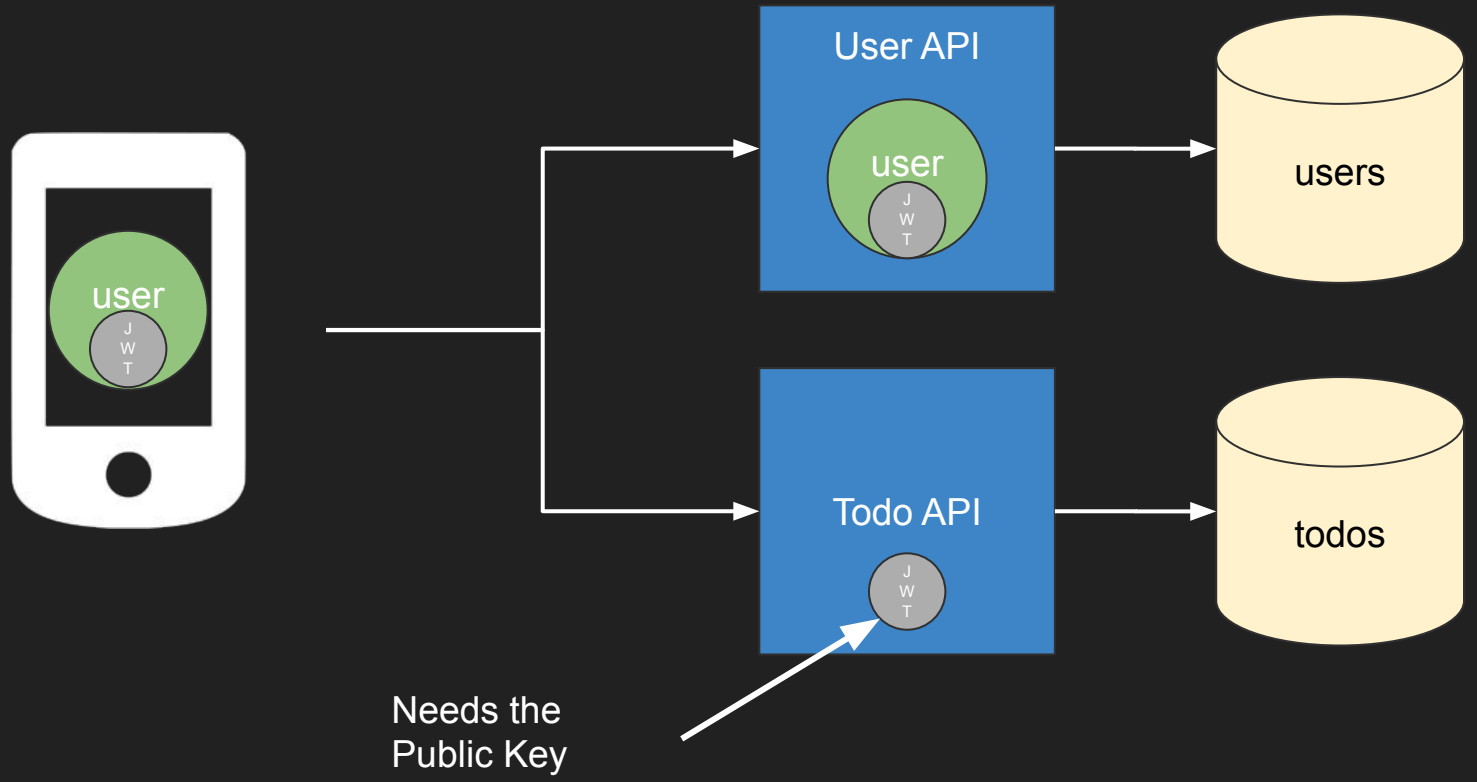
Why Not?

- Slower

Why Not?

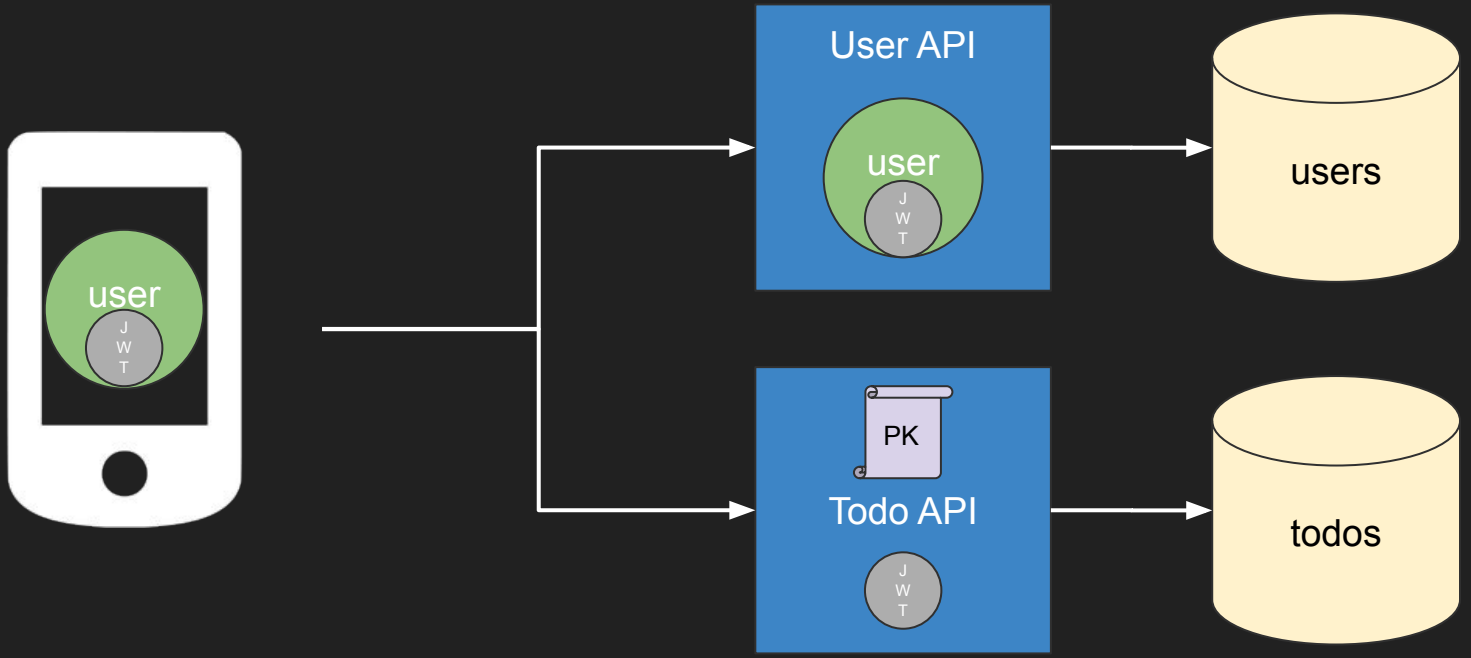
- Slower
- More complex

Sharing Public Keys


user
J
W
T
User API
user
J
W
T
users
Todo API
J
W
T
todos
Needs the
Public Key

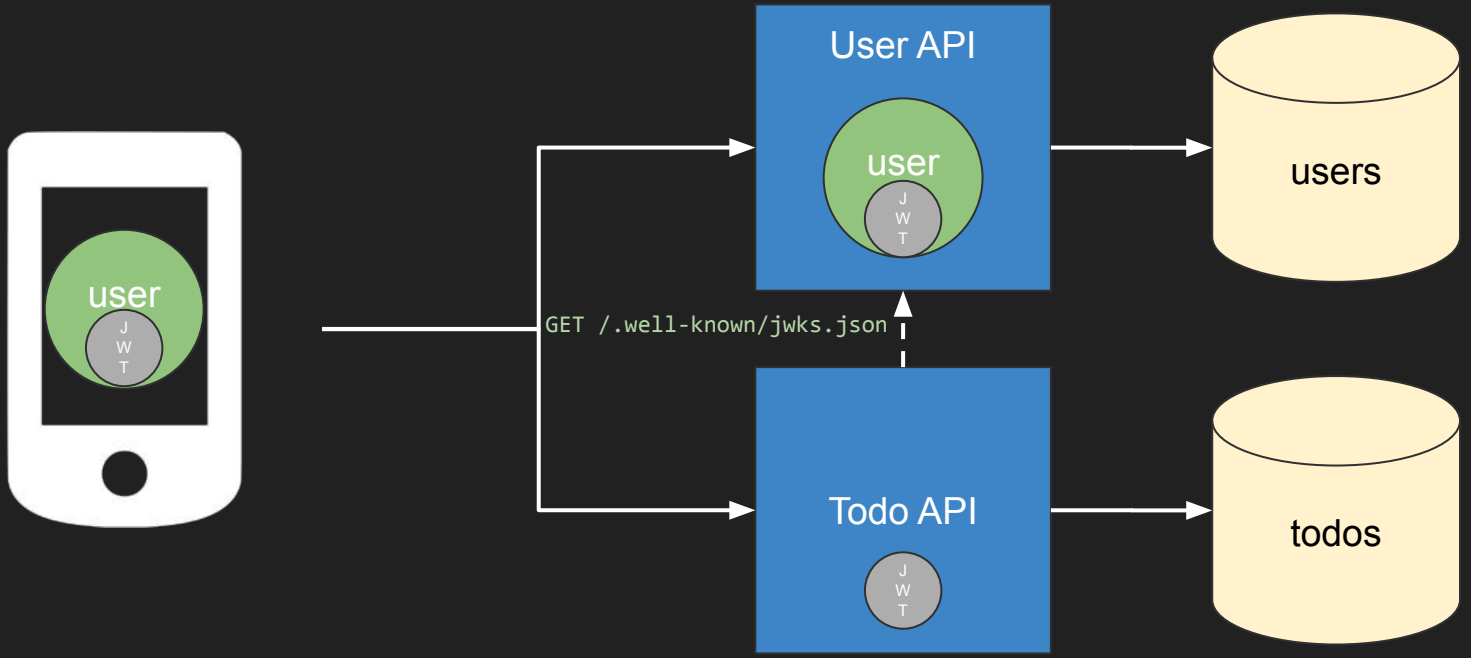
Sharing Public Keys

- Distribute the keys


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

Sharing Public Keys

- Distribute the keys
- JWKS, another standard (RFC 7517)


user
J
W
T
GET /.well-known/jwks.json
User API
user
J
W
T
users
Todo API
J
W
T
todos

```
{
  "keys" : [
    {
      "alg" : "RS256",
      "e" : "AQAB",
      "kid" : "uk0PWf8KkTKiJqWwrNjn16QoKKI",
      "kty" : "RSA",
      "n" :
"2xzTUGNSpIeNcvICSlFlhver42dR9CWYePEoLk6ncuk1nKLzwOr-hy3W2rmkG-x_DaVMVBT5jimClL_k7Fu5xlscexpmNTo3lK_fqWv_qh
nOONSaX0ETqWSrS9MXWnJcPTZkA37ZAwhGKaz8zzSF3Jh_fULWnFHgJxCLBNYmopnvVAv_erJR0wjX9imMpMsBh3w8O6RyN8ghhlkJ0q4JK
yaauF-xk8nLwIAvdWiPWNpzJ57oXfHUXesbLCfLIuM3f_suh_RaZn7uC0jEO1uG23htOqMb1M0TW5uk8pAxdhVKYbKsPR8CMOUc1je8wDo2
w4y4gY_1koST3xnA6IRhBQ",
      "use" : "sig",
      "x5c" : [
"MIICuDCCAaCgAwIBAQIQMcWR+VPwTieC06b3Fn6XbzANBgkqhkiG9w0BAQsFADAYMRYwFAYDVQQDEw1mdXNpb25hdXRoLmlvMB4XDTIxMD
UyNDE5NDU1OVoXDTMxMDUyNDE5NDU1OVowGDEWMBQGA1UEAxMNZnVzaW9uYXV0aC5pbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCg
gEBANsc01BjUqSHjXLyAkpRZYb3q+NnUfQlmHjxKC5Op3LpNZyi88Dq/oct1tq5pBvsfw2lTFQU+Y4pgpS/5OxbucZbHHsaZjU6N5Sv36lr
/6oZzjjUml9BE6lkq0vTF1pyXD02ZAN+2QMIRims/M80hdyYf31C1pxR4CcQiwTWJqKZ71QL/3qyUdMI1/YpjKTLAYd8PDukcjfIIYZZCdK
uCSsmmrhfsZPJy8CAL3Voj1jacyee6F3x1F3rGywnyyLjN3/7Lof0WmZ+7gtIxDtbhtt4bTqjG9TNE1ubpPKQMXYVSmGyrD0fAjDlHNY3vM
A6NsOMuIGP9ZKEk98ZwOiEYQUCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAcj/NIIltfhyP9zsleVn7N/QRavfKA1SBTwtlPMVezuRIX+S3j
zxJb/ot47TBD5WFNY5y5AOkWHQFNkVtuPjUYmKTAqJd0+kVur77tLKzour6wjOp2QgKzG3IGxQnK903JkFlyWF4vSJuOpH8WymJ1jq1gD5z
Jjz2NXqch+gBIp5Kscr2tj2hg2BGmq5v7+5pz2jYHosarj4sJwGsLqk1j479wK1iaMjdBVCMuq/QS9yF0sG0STsjbceyGzFWbknmZdfNup6
C4a0m8paeb9mlgFfIx9qljuNInpc9QZWeRymNJ5spX0WYVRLu7ULrLDXr8xUupjCSMV95yI1XF6tMkw=="
      ],
      "x5t" : "uk0PWf8KkTKiJqWwrNjn16QoKKI",
      "x5t#S256" : "UW-4zdFS6YR9g4Vhl3T3xLwfQ_S1aLFh2x4VBvK1sbY"
    }
  ]
}
```

```
{
    "alg" : "RS256",
    "kid" : "uk0PWf8KkTKiJqWwrNjn16QoKKI",
    "x5c" : [
            "MIICuDCCAaCgAwIBAQIQMcWR+VPwTi..."
    ]
    // ...
}
```

Which Matches

```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "uk0PWf8KkTKiJqWwrNjn16QoKKI"
}
```

Refresh
Tokens

Lifetimes

- JWTs are meant to be short lived (minutes)

Lifetimes

- JWTs are meant to be short lived (minutes)
- Refresh tokens are long lived (days or months)

Lifetimes

- JWTs are meant to be short lived (minutes)
- Refresh tokens are long lived (days or months)
- Refresh tokens can be used to create new JWTs

Why?

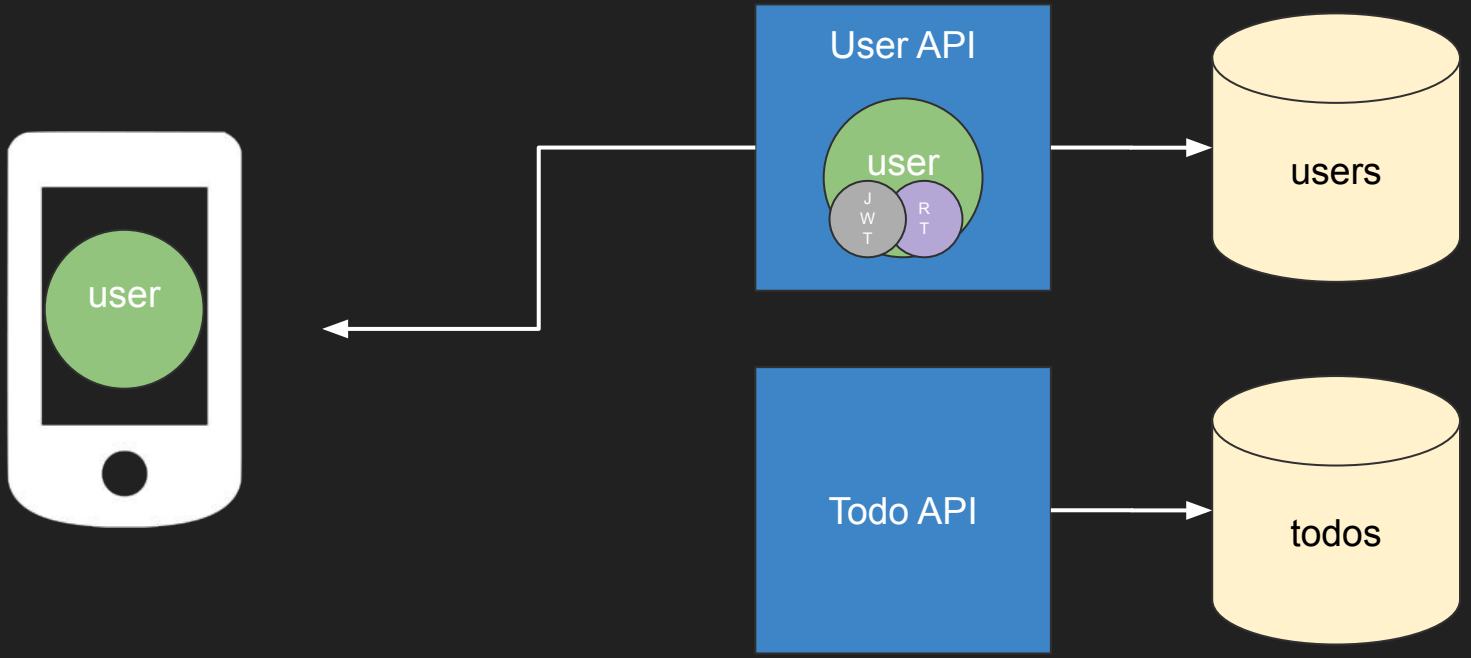
- Compromise between two unsavory options

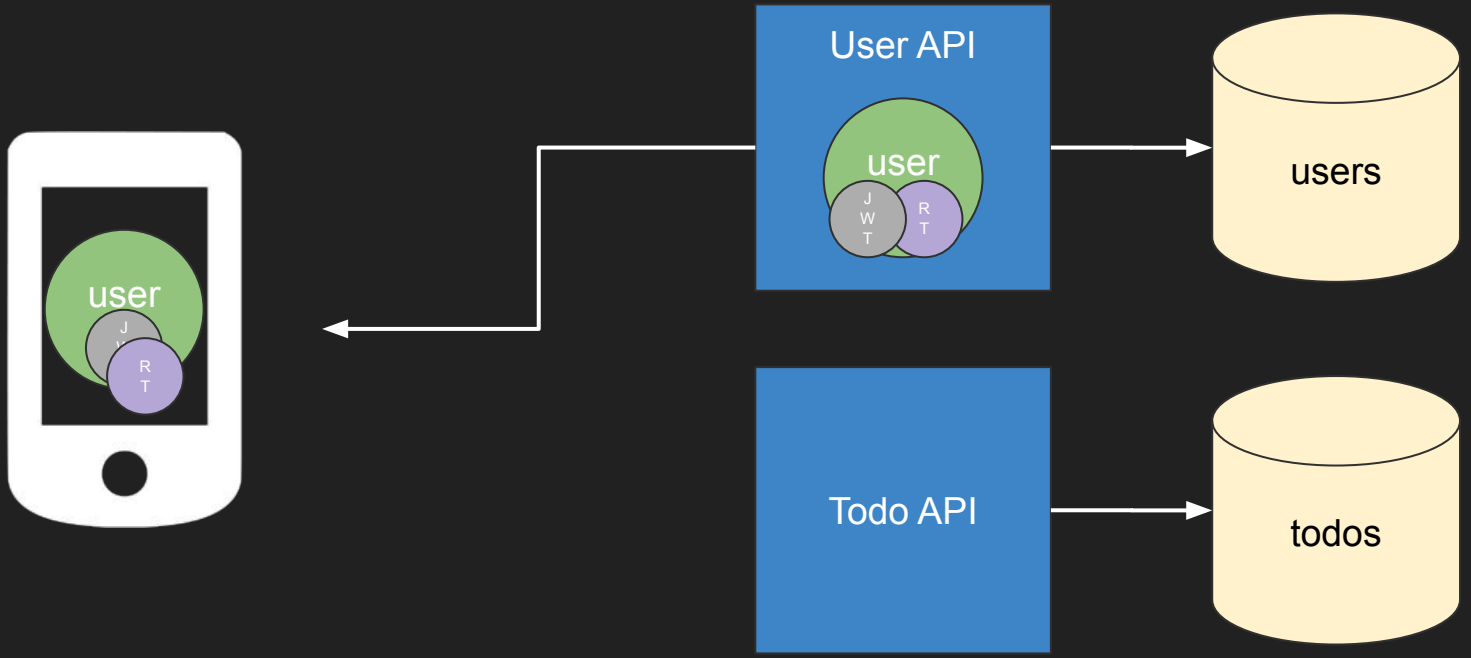
Why?

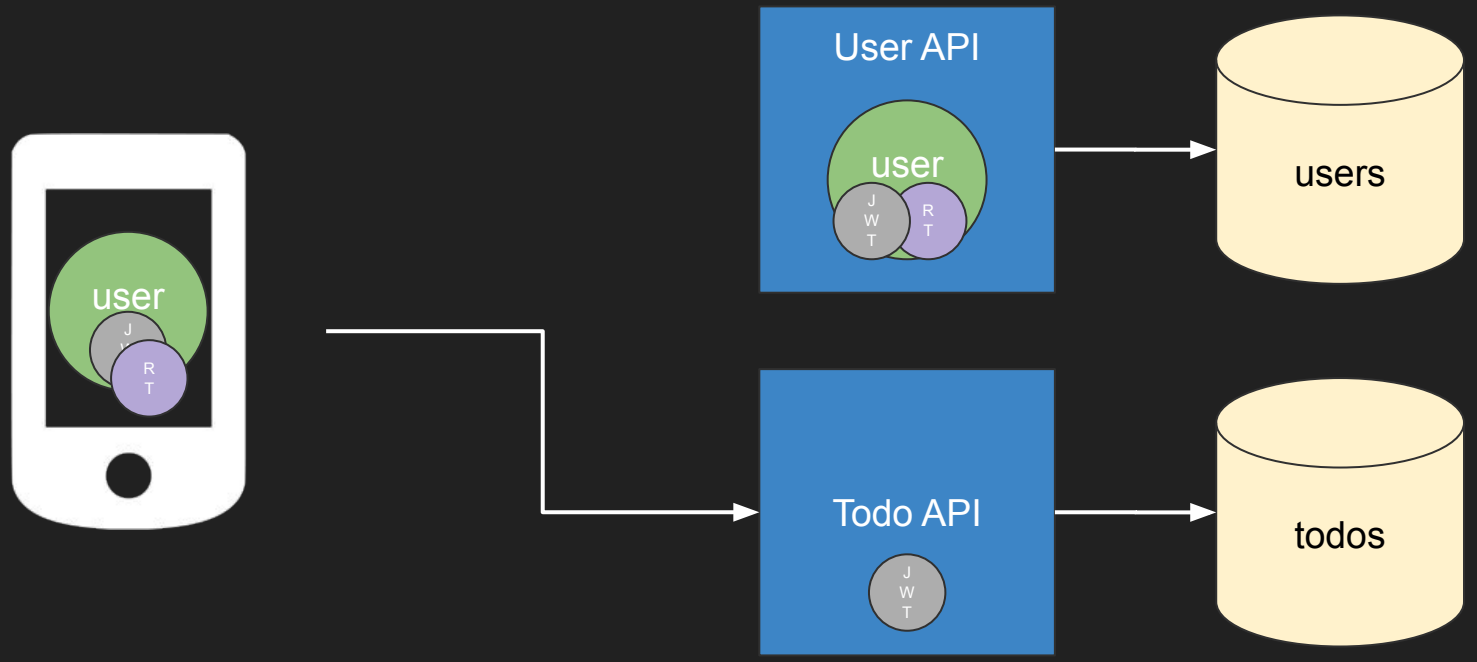
- Compromise between two unsavory options
- User authenticating regularly
  - User hassle

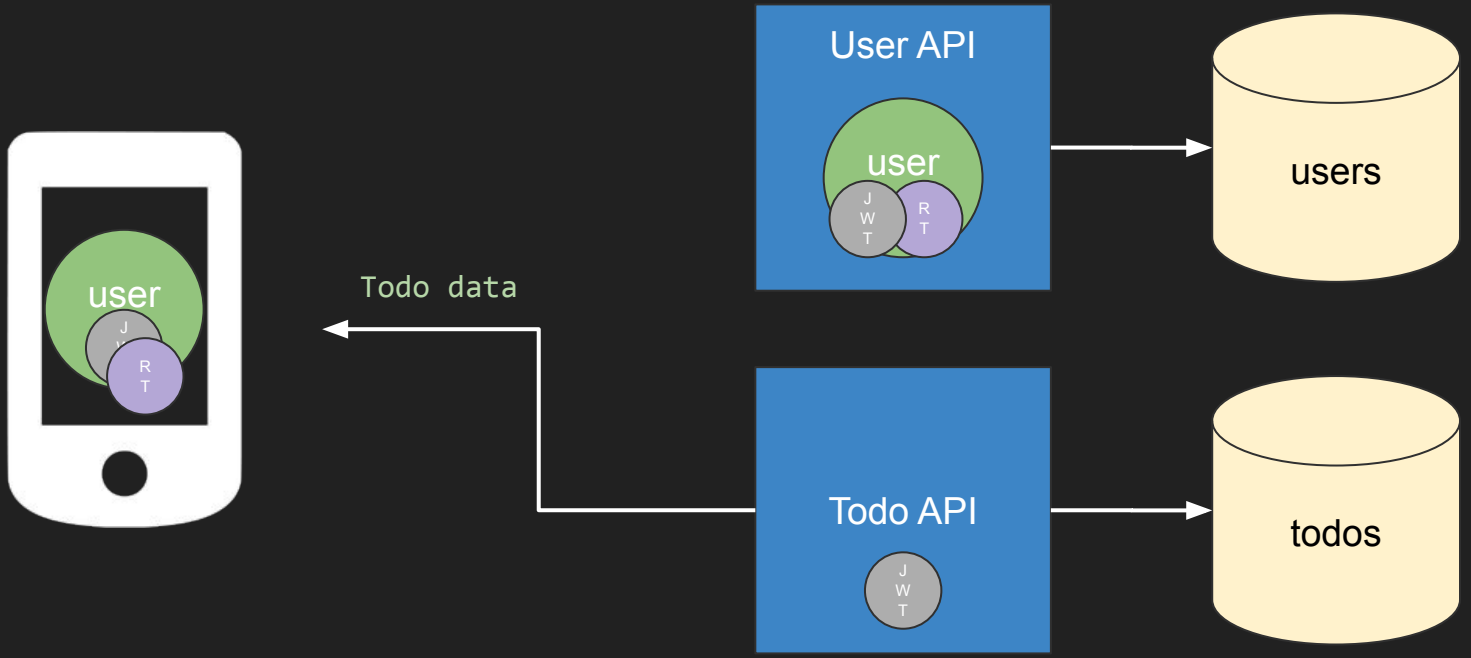
Why?

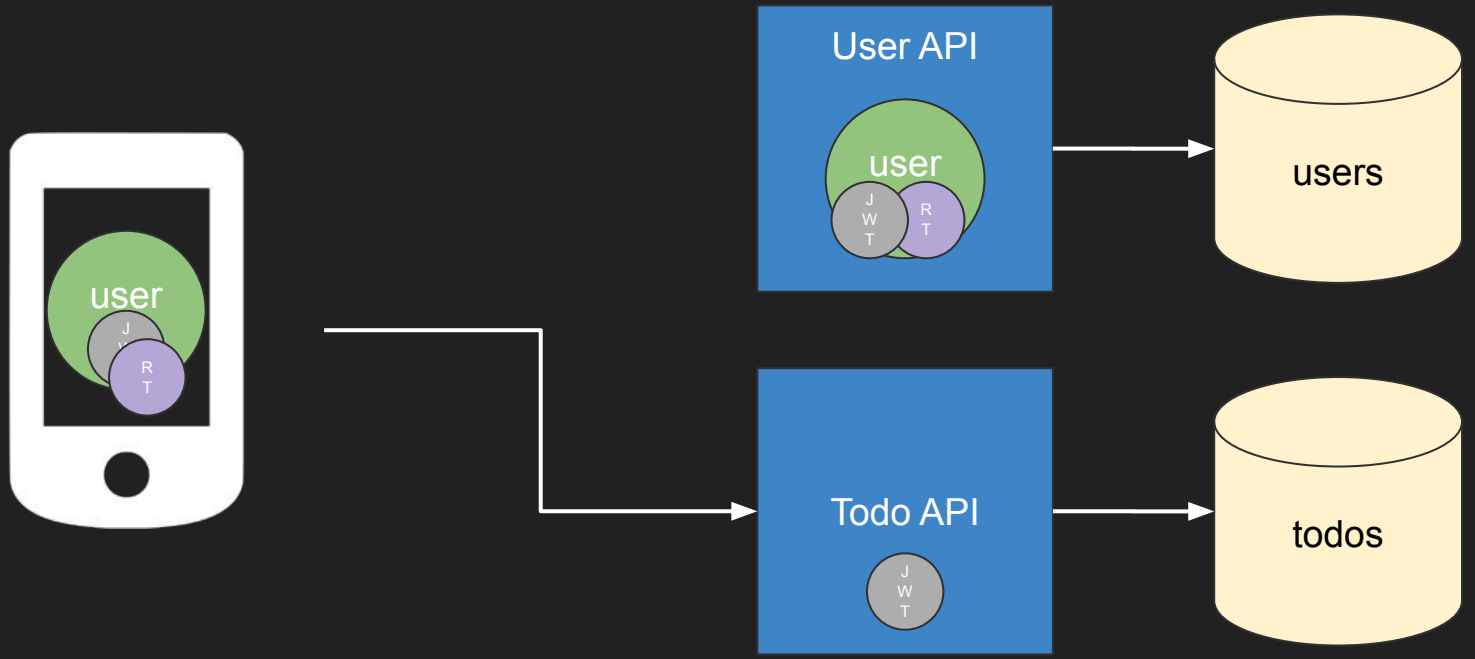
- Compromise between two unsavory options
- User authenticating regularly
  - User hassle
- Long lived JWTs
  - Security risk

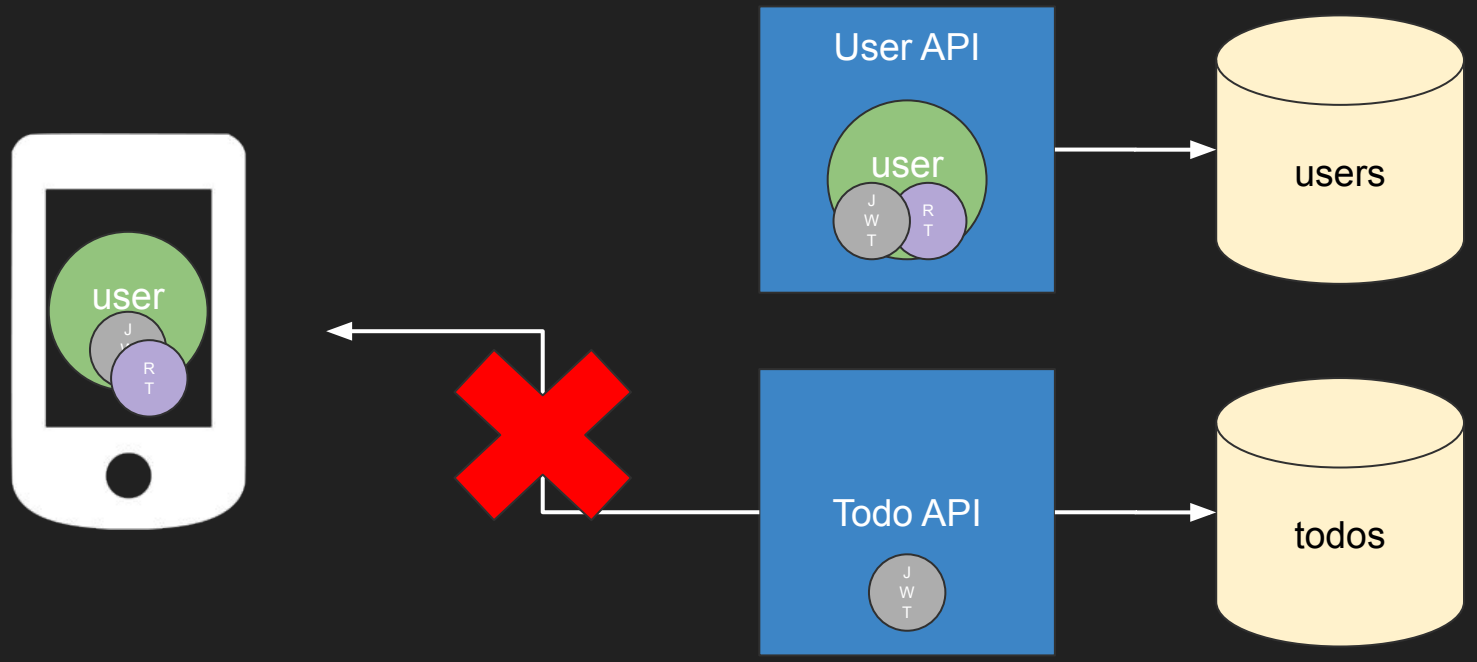

User API
user
J
W
T
R
T
users
user
Todo API
todos

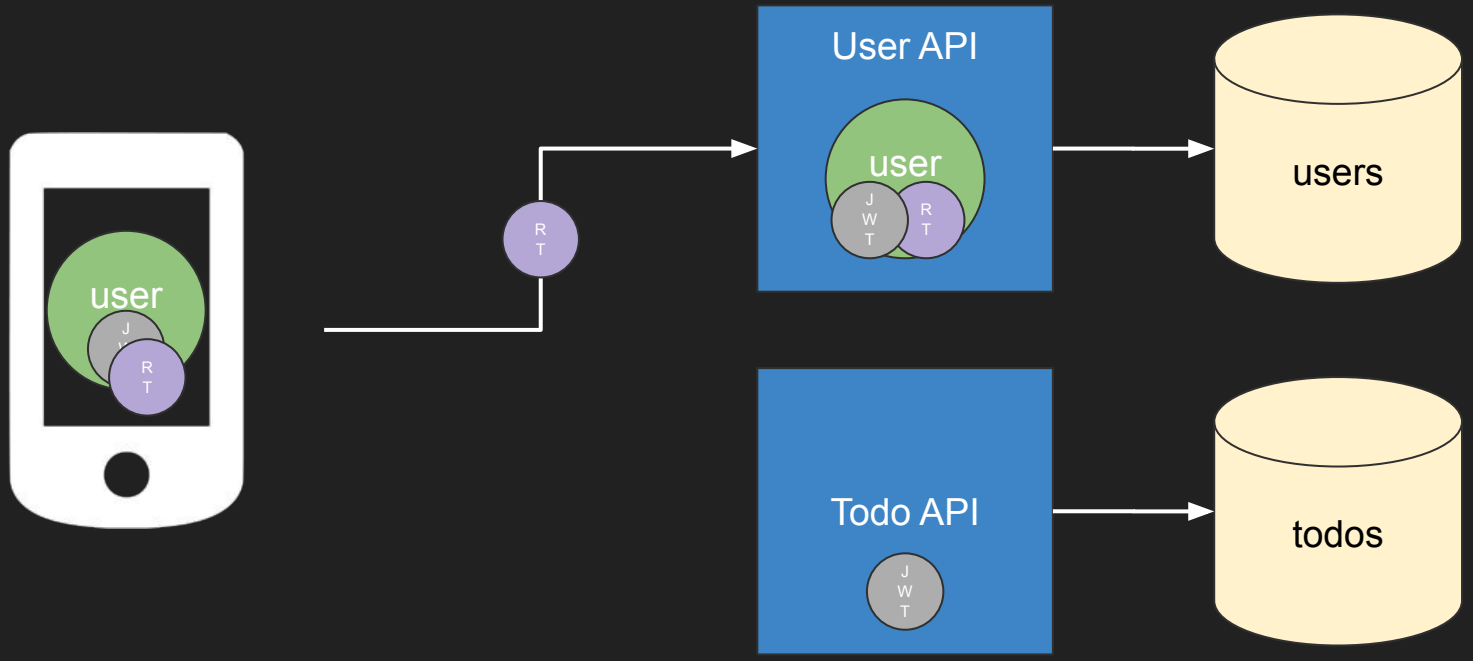

User API
user
J
W
T
R
T
users
user
J
R
T
Todo API
todos

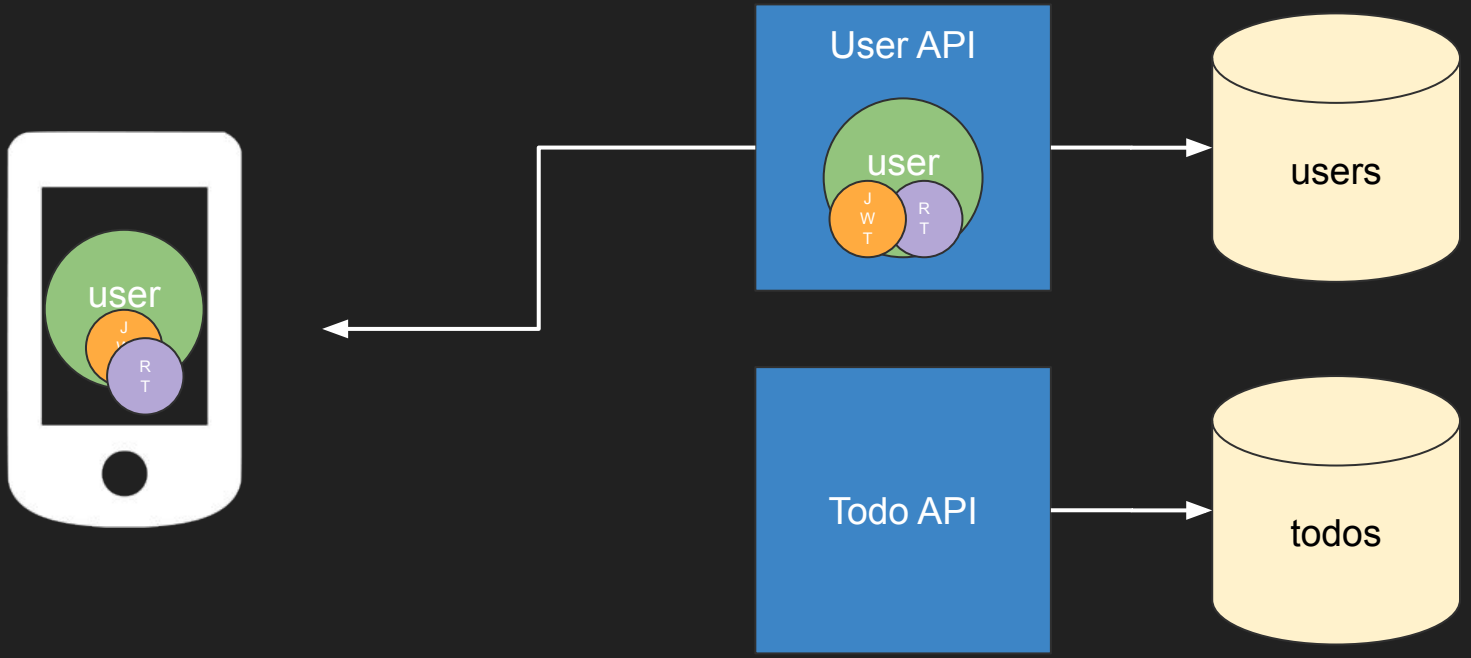

User API
user
J
W
T
R
T
users
user
J
R
T
Todo API
J
W
T
todos

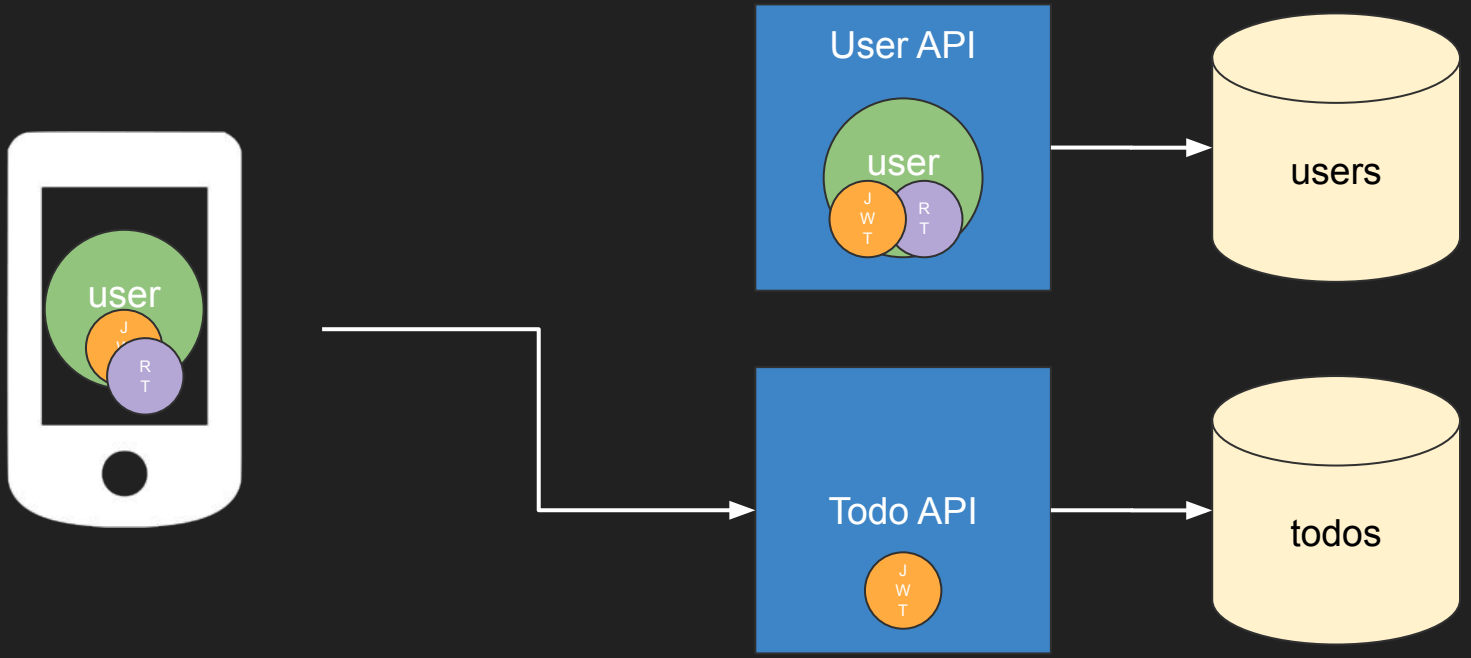

User API
user
J
W
T
R
T
users
user
J
R
T
Todo data
Todo API
J
W
T
todos

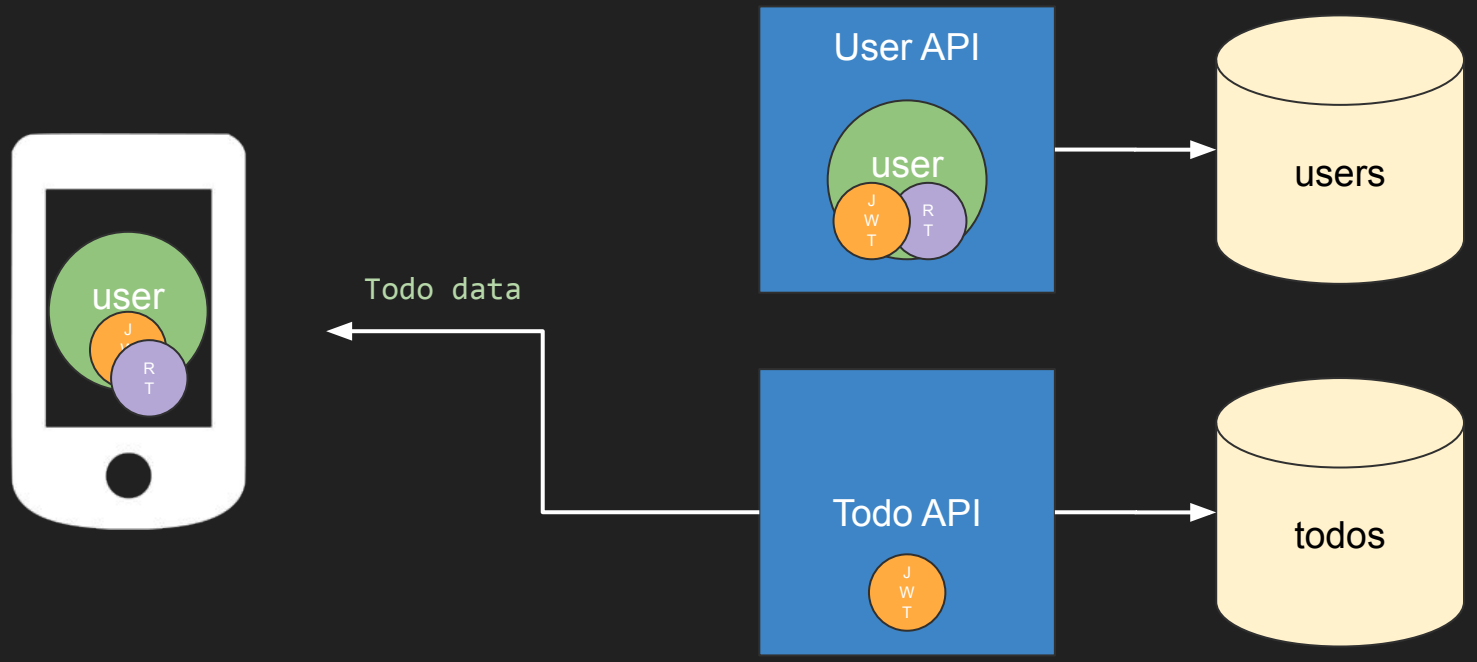

User API
user
J
W
T
R
T
users
user
J
R
T
Todo API
J
W
T
todos


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


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

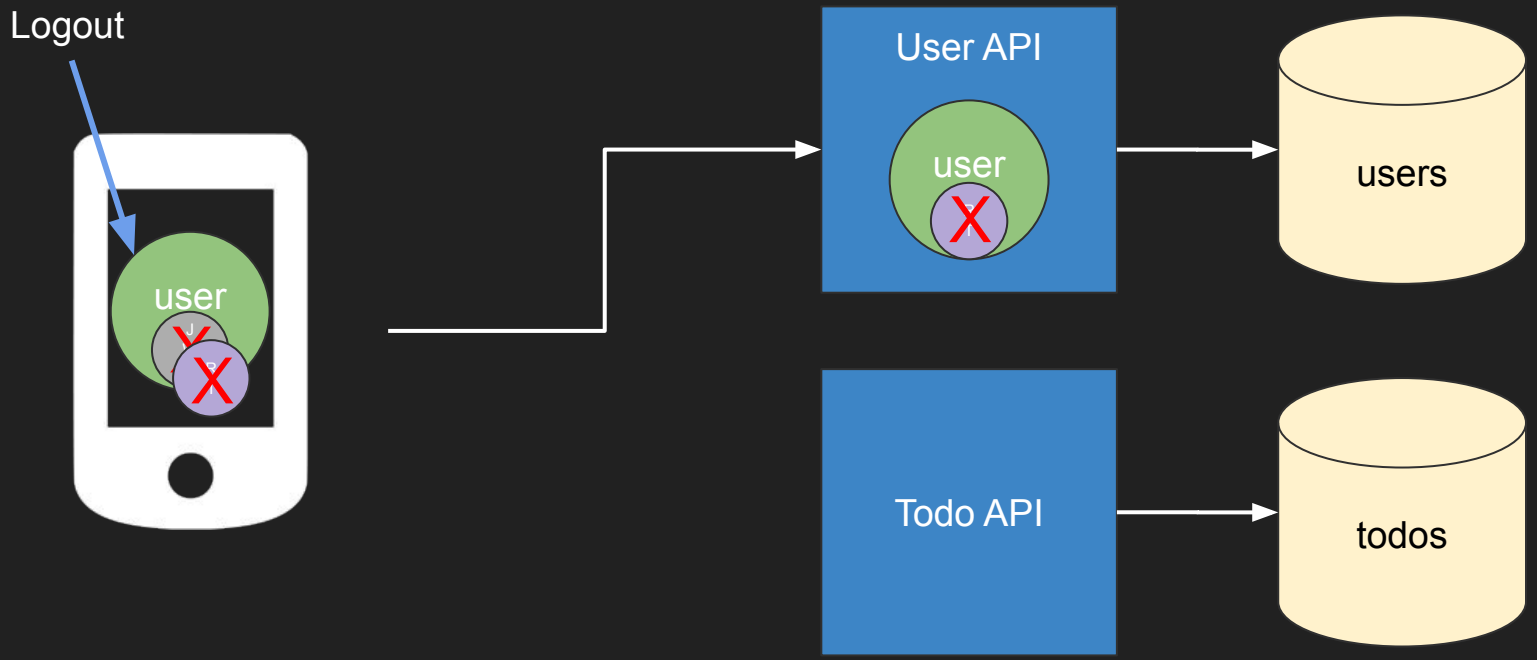

User API
user
J
W
T
R
T
users
user
J
R
T
Todo API
todos


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


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

Logout

- Revoking refresh token


Logout
user
User API
user
users
Todo API
todos

JWT Revocation

Can You Do It?

Kinda

- You have a few options

Kinda

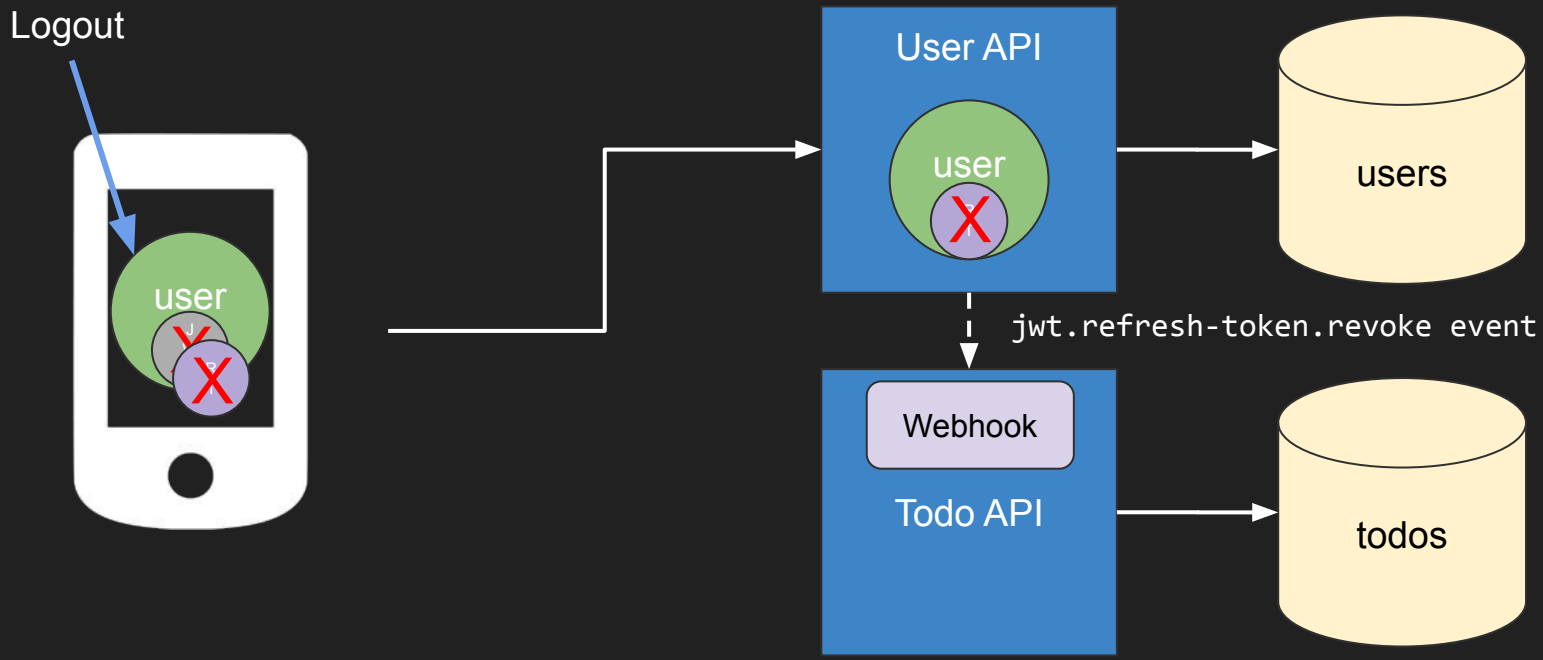
- You have a few options
  - Rotate keys

Kinda

- You have a few options
  - Rotate keys
  - Really short lifetimes

Kinda

- You have a few options
  - Rotate keys
  - Really short lifetimes
  - Deny list


Logout
user
User API
user
users
jwt.refresh-token.revoke event
Webhook
Todo API
todos

In Conclusion

Thanks & Questions

- Contact me for more info:
  - dan@fusionauth.io
  - https://twitter.com/mooreds
  - https://fusionauth.io
- Links
  - https://fusionauth.io/learn/expert-advice/dev-tools/jwt-debugger
  - https://github.com/FusionAuth/fusionauth-example-javascript-jwt/
  - https://fusionauth.io/learn/expert-advice/tokens/building-a-secure-jwt
  - https://fusionauth.io/learn/expert-advice/tokens/anatomy-of-jwt
  - FusionAuth Community Edition (free!): https://fusionauth.io/download
  -