

# OAuth

## Daniel Tull

# Questions

# “Why??”

To allow users to authenticate your app without giving you their login details

# “Why??”

It provides a generic method for API authentication

“When is it going away?”

In a long long time

Unfortunately

“Is it true that OAuth led to your  
recent killing spree?”

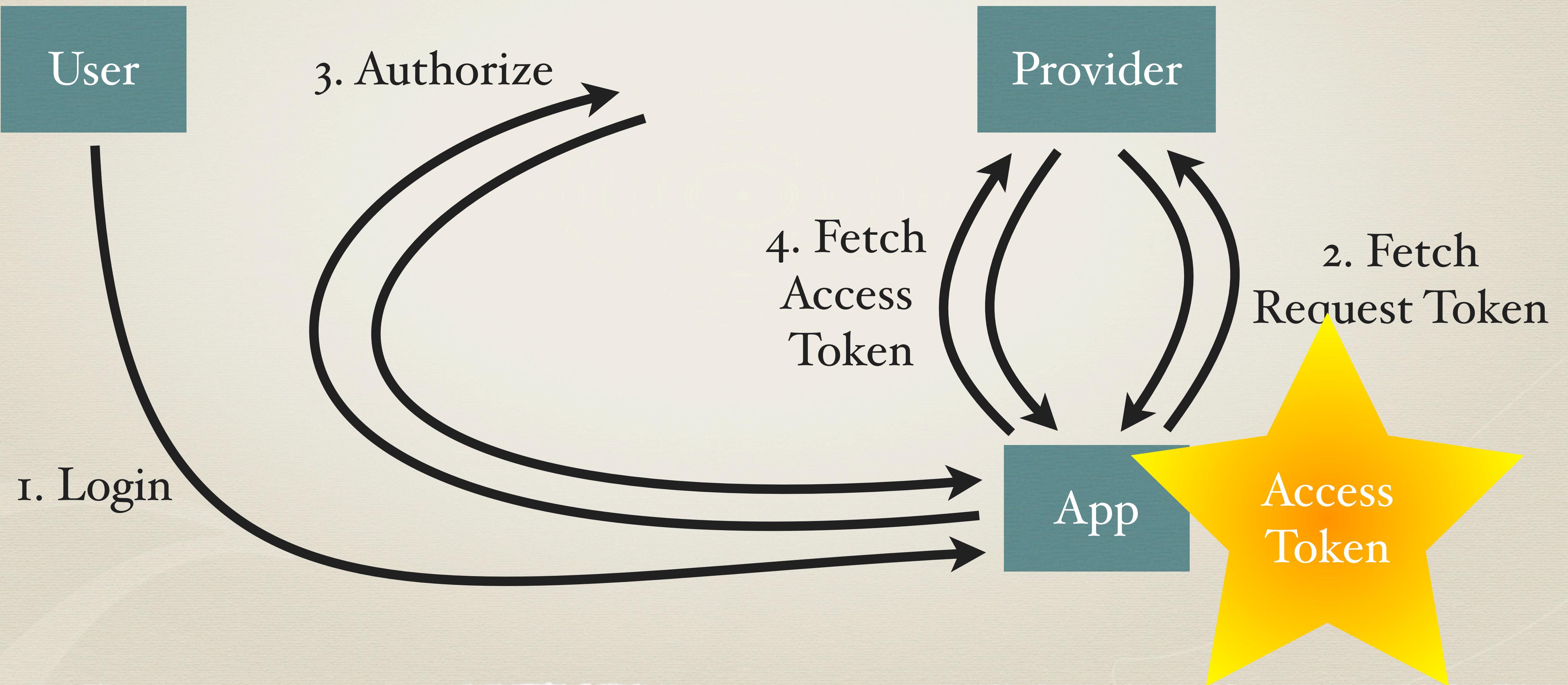
Heh heh heh heh

No

“Is it true that OAuth 2.0 is \*&\$%  
(like the ex-lead author says)?”

Yes

# OAuth Flow



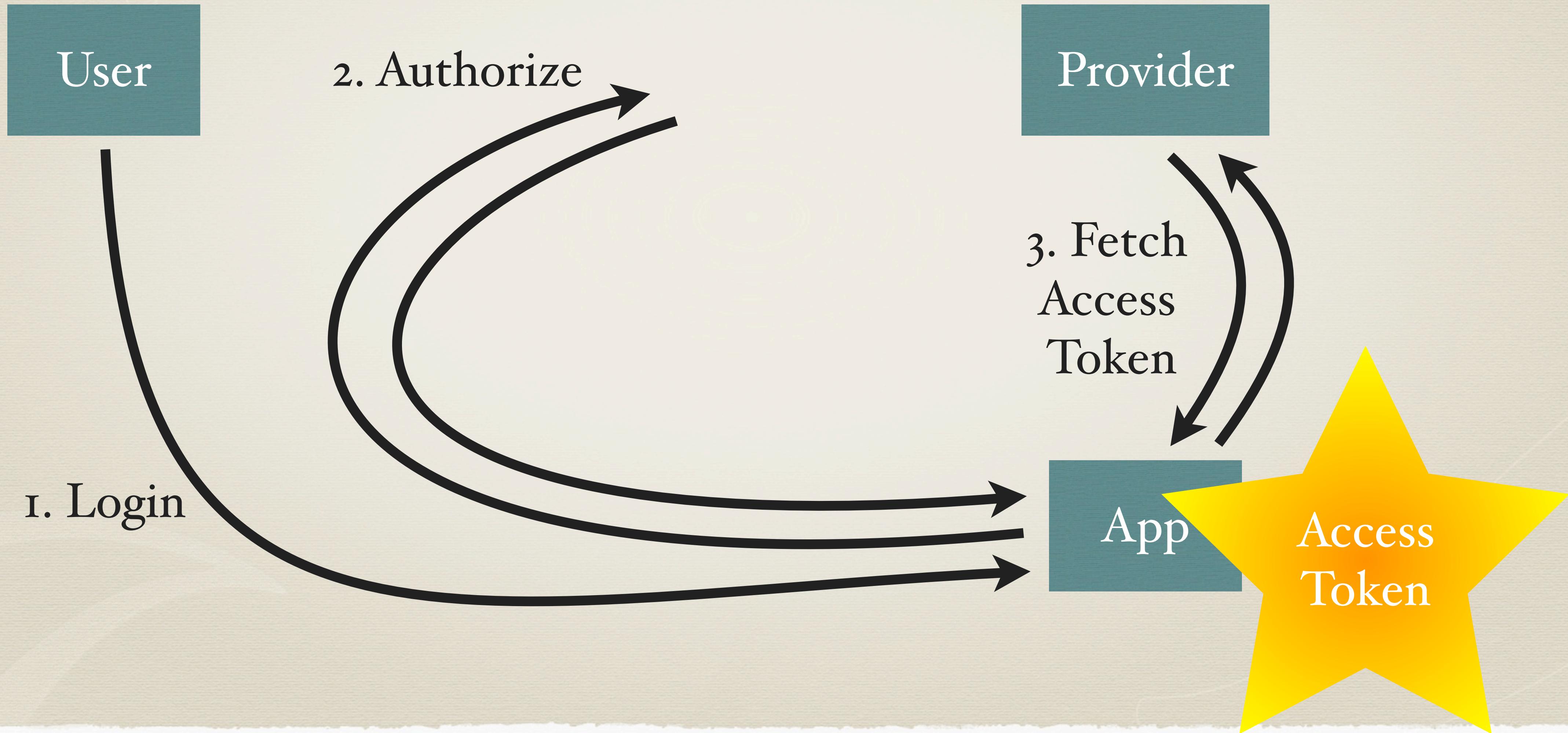
# OAuth

Once you figure out the correct signing process,  
it's easy

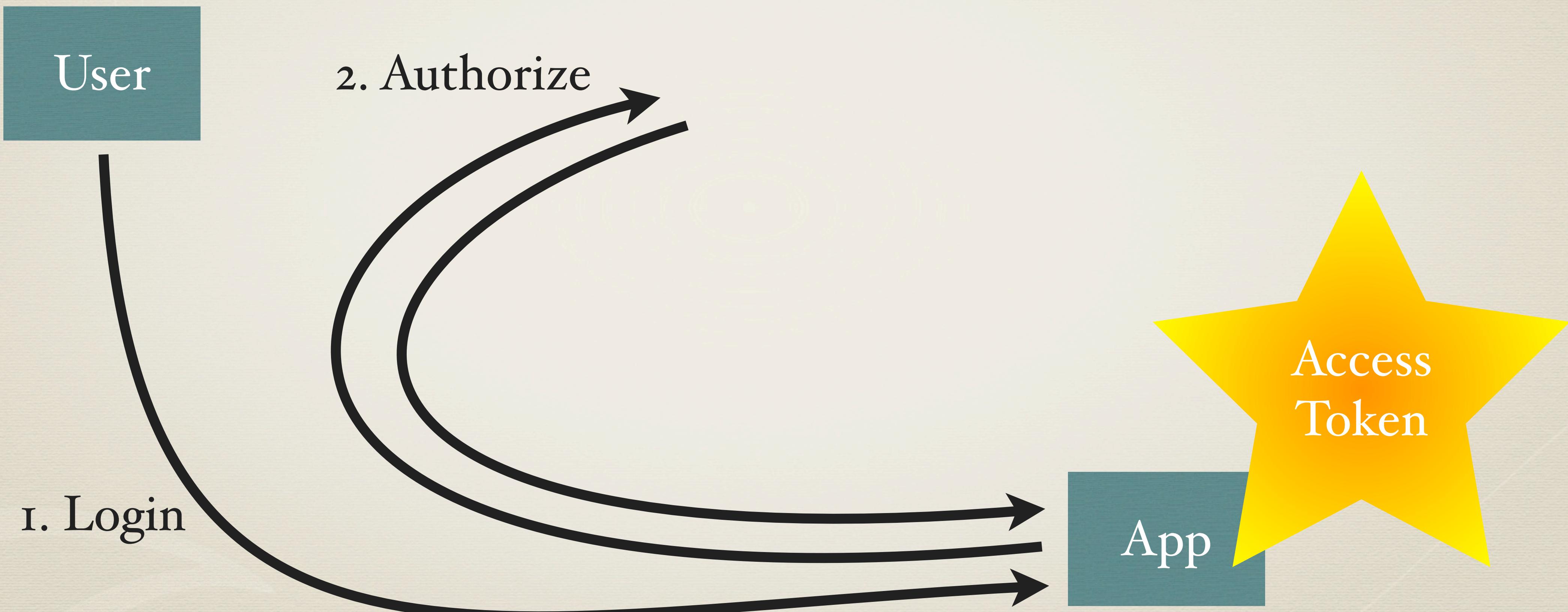
# OAuth

It's generally the same for every provider

# OAuth 2.0 Flow



# Another OAuth 2.0 Flow



# OAuth 2.0

The “standard” is too open

# OAuth 2.0

There's many subtle differences in the ways a provider could implement it

# OAuth 1.0 vs OAuth 2.0

	OAuth 1.0	OAuth 2.0
Implementation	Tricky	Easy
Debugging	Frustrating	Less frustrating
Reusability	Everywhere	???
Security on iOS	Requires secret keys	Sometimes doesn't
User Experience	Horrid	Horrid

“So you’ve made an open source library to solve all this, right?”

Yes

“So you’ve made an open source  
library to solve all this, right?”

DCTAuth

<http://github.com/danielctull/DCTAuth>

“And you learnt some things?”

Twitter requires the callback URL to have a host

dctauth://host

(Actually this makes sense)

“And you learnt some things?”

Facebook requires the callback URL to have a  
scheme of fb[app id]://

fb4519042343://authorize/

“And you learnt some things?”

Facebook requires the callback URL to have a host of authorize **with** a trailing slash

fb4519042343://authorize/

“And you learnt some things?”

Oh, except when calling out to the Facebook app  
for authorization, then it doesn't want the slash

`fb4519042343://authorize`

Daniel Tull  
[danieltull.co.uk](http://danieltull.co.uk)