Federated Logins with OAuth 2.0 & OpenID Connect

AppSec New Zealand 2021

Thank You to Our Sponsors and Hosts!

OWASP NEW ZEALAND owasp.org.nz

THE UNIVERSITY OF AUCKLAND Te Whare Wānanga o Tāmaki Makaurau NEW ZEALAND

DATACOM

QUANTUM SECURITY

ATTACK DEFEND Group Company

aura INFORMATION SECURITY POWERED BY KORDIA

INSOMNIA SECURITY SPECIALISTS :: REST SECURED

myob

Pulse Security www.pulsesecurity.co.nz

SECURE CODE WARRIOR

VOCUS

ZX SECURITY

flux

internetnz

LATERAL SECURITY

secInit security initiative

SEQA™ Information Security

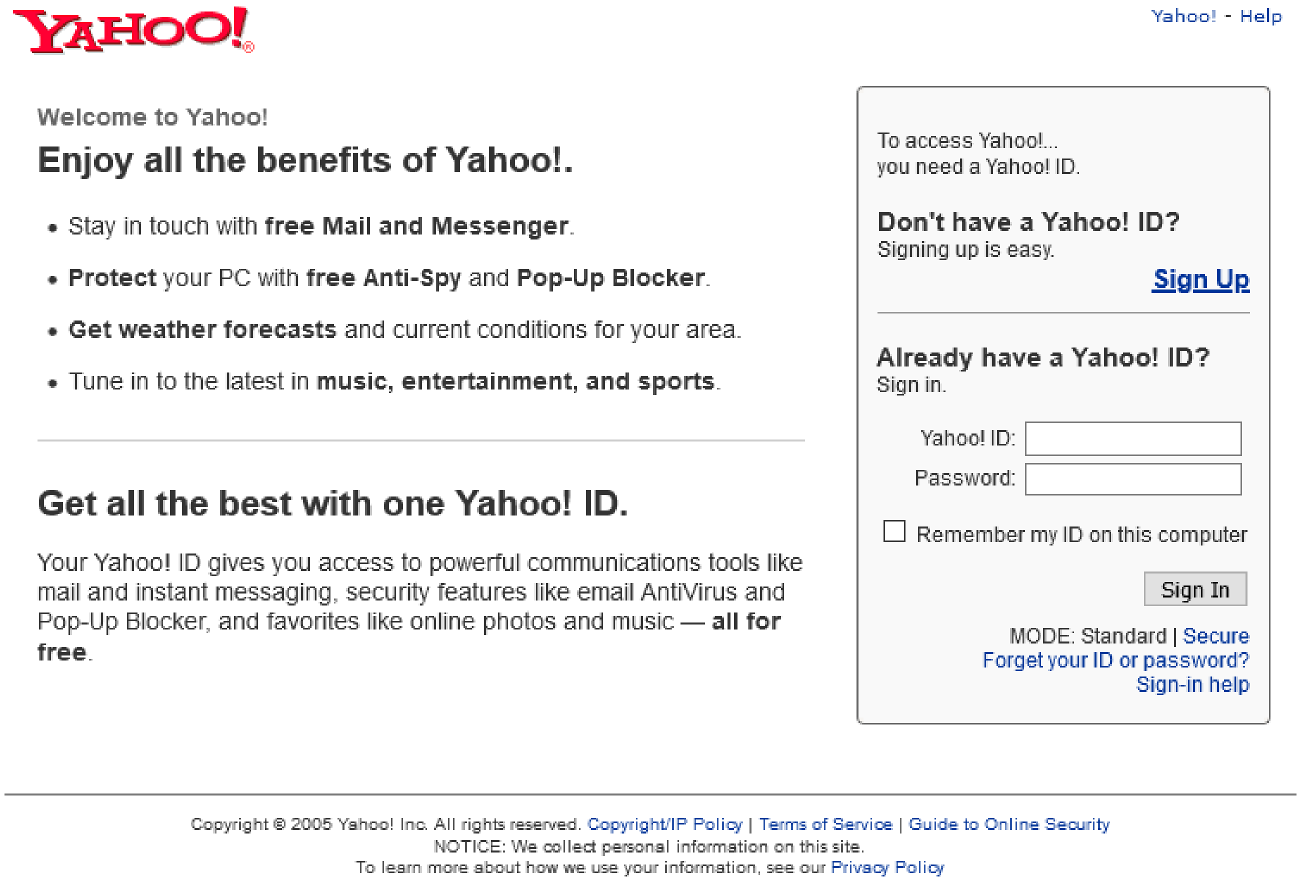
Without them, this Conference couldn't happen

Also, thank you!

#whoami

- Matt Cotterell/TC
- ZX Security – Security Consultant
- Auth stuff, Cloud stuff, .NET stuff
- Ridiculous stock photos are my jam

Traditionally, login forms looked like this:



Yahoo! - Help

Welcome to Yahoo!

Enjoy all the benefits of Yahoo!.

- Stay in touch with **free Mail and Messenger**.
- **Protect** your PC with **free Anti-Spy** and **Pop-Up Blocker**.
- **Get weather forecasts** and current conditions for your area.
- Tune in to the latest in **music, entertainment, and sports**.

Get all the best with one Yahoo! ID.

Your Yahoo! ID gives you access to powerful communications tools like mail and instant messaging, security features like email AntiVirus and Pop-Up Blocker, and favorites like online photos and music — **all for free**.

To access Yahoo!...
you need a Yahoo! ID.

Don't have a Yahoo! ID?
Signing up is easy.

Sign Up

Already have a Yahoo! ID?
Sign in.

Yahoo! ID:	
Password:	

☐ Remember my ID on this computer

Sign In

MODE: Standard | Secure
Forget your ID or password?
Sign-in help

Copyright © 2005 Yahoo! Inc. All rights reserved. Copyright/IP Policy | Terms of Service | Guide to Online Security
NOTICE: We collect personal information on this site.
To learn more about how we use your information, see our Privacy Policy

Local Authentication

- Password between user and service
  - Never given to any other service
- User’s stuff was on the same service
  - Other services couldn’t access this stuff

Passwords kind of *suck*…

- Storing them safely is hard
  - Service has to hash it, salt it, iterate it, etc
  - Users have to use a password manager (or likely, just reus
- Can’t give this to some other service to use
  - Grants “unlimited access” to account
  - Resetting password breaks access for all services

Fast forward to 2021…

- Data breaches keep leaking our passwords
- Too many passwords, we want less things to remember
- We want services to work together
- We want to control our data
    - Share some things, not everything
    - Revoke this later

Introducing: OpenID Connect, OAuth 2.0 and JWT

OpenID Connect
Authentication

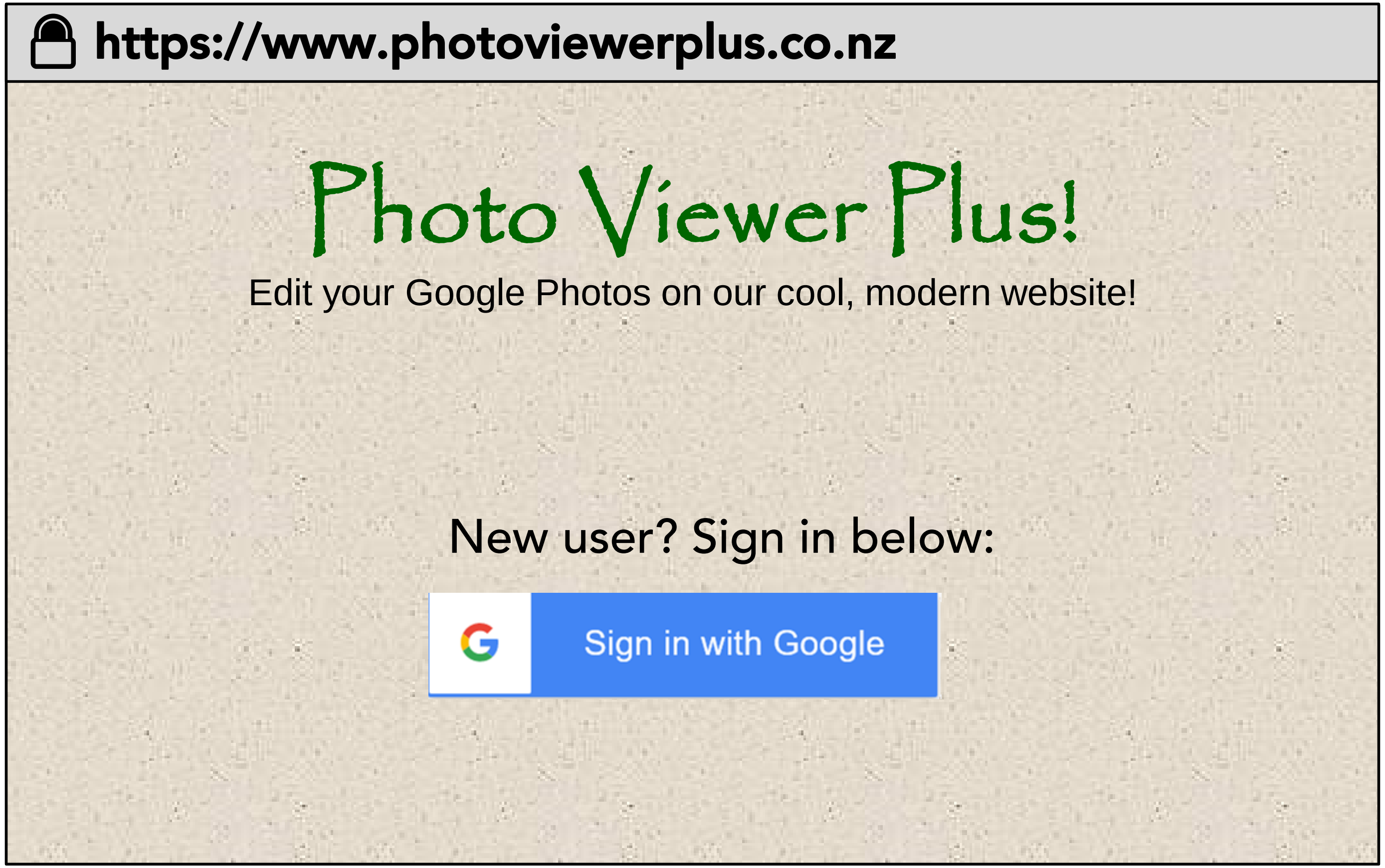
- Log in with an external provider
- Tells services who you are
- Choose to share additional information about you
- An "extension" of OAuth 2.0

OAuth 2.0
Authorization

- Lets services access some of your data
- Works in conjunction with OpenID Connect

JWT
Token Format

- Given to the service
- Used to "prove" your identity
- (Often) used as a "temporary token" to access your resources

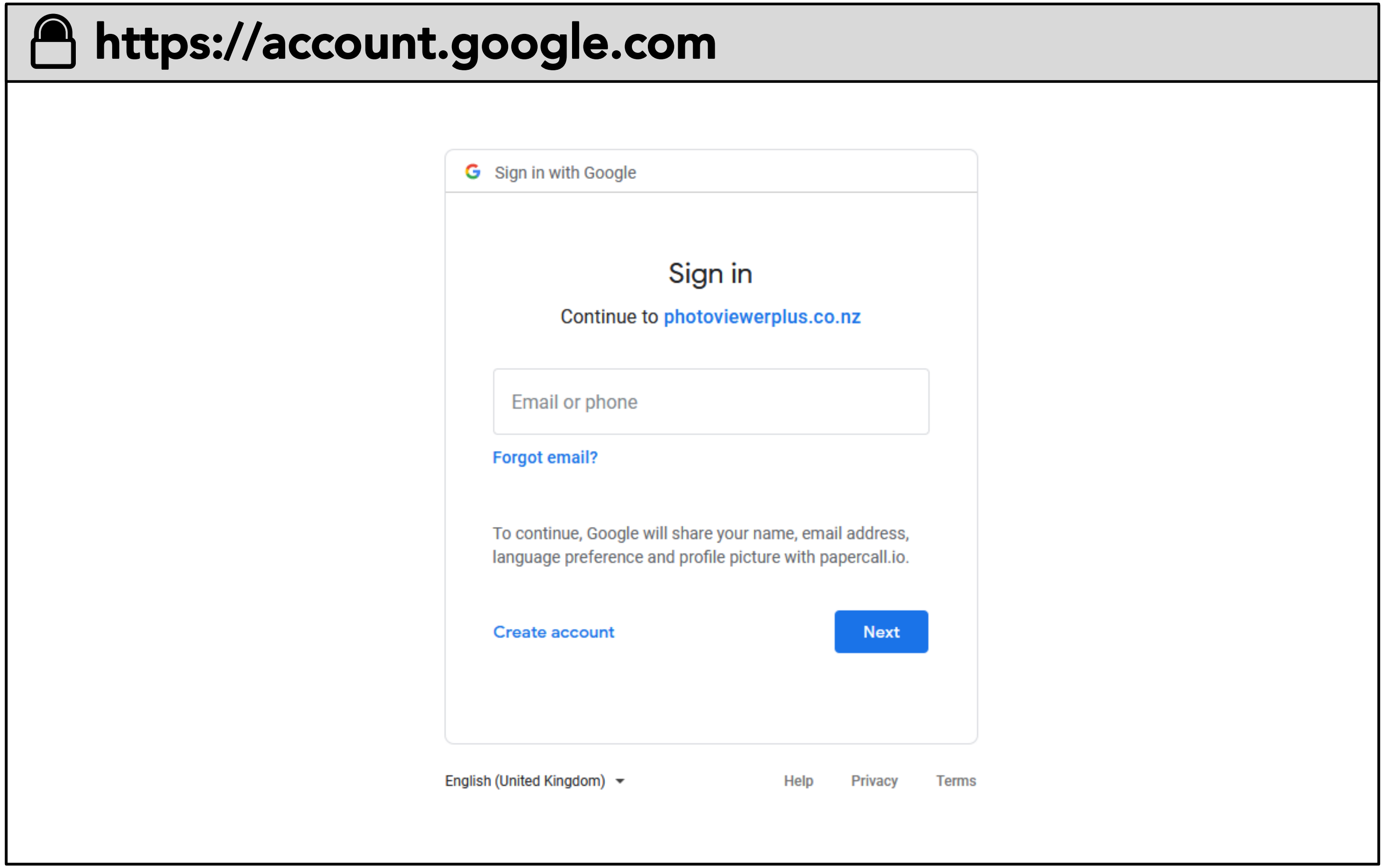


https://www.photoviewerplus.co.nz

Photo Viewer Plus!

Edit your Google Photos on our cool, modern website!

New user? Sign in below:


https://account.google.com

Sign in with Google

Sign in

Continue to photoviewerplus.co.nz

Email or phone

Forgot email?

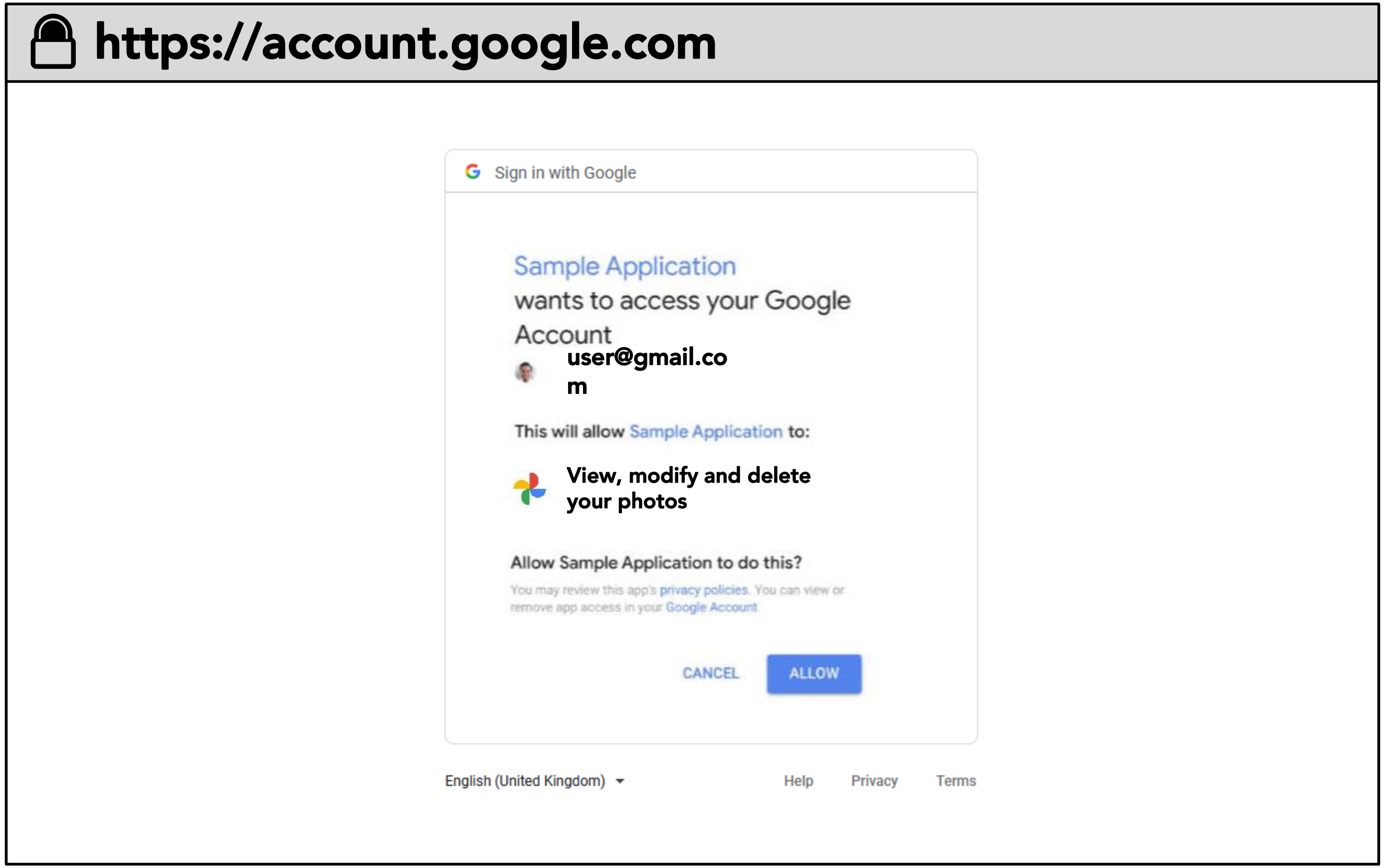
To continue, Google will share your name, email address, language preference and profile picture with papercall.io.

Create account

Next

English (United Kingdom)

Help Privacy Terms



https://account.google.com

Sign in with Google

Sample Application
wants to access your Google Account

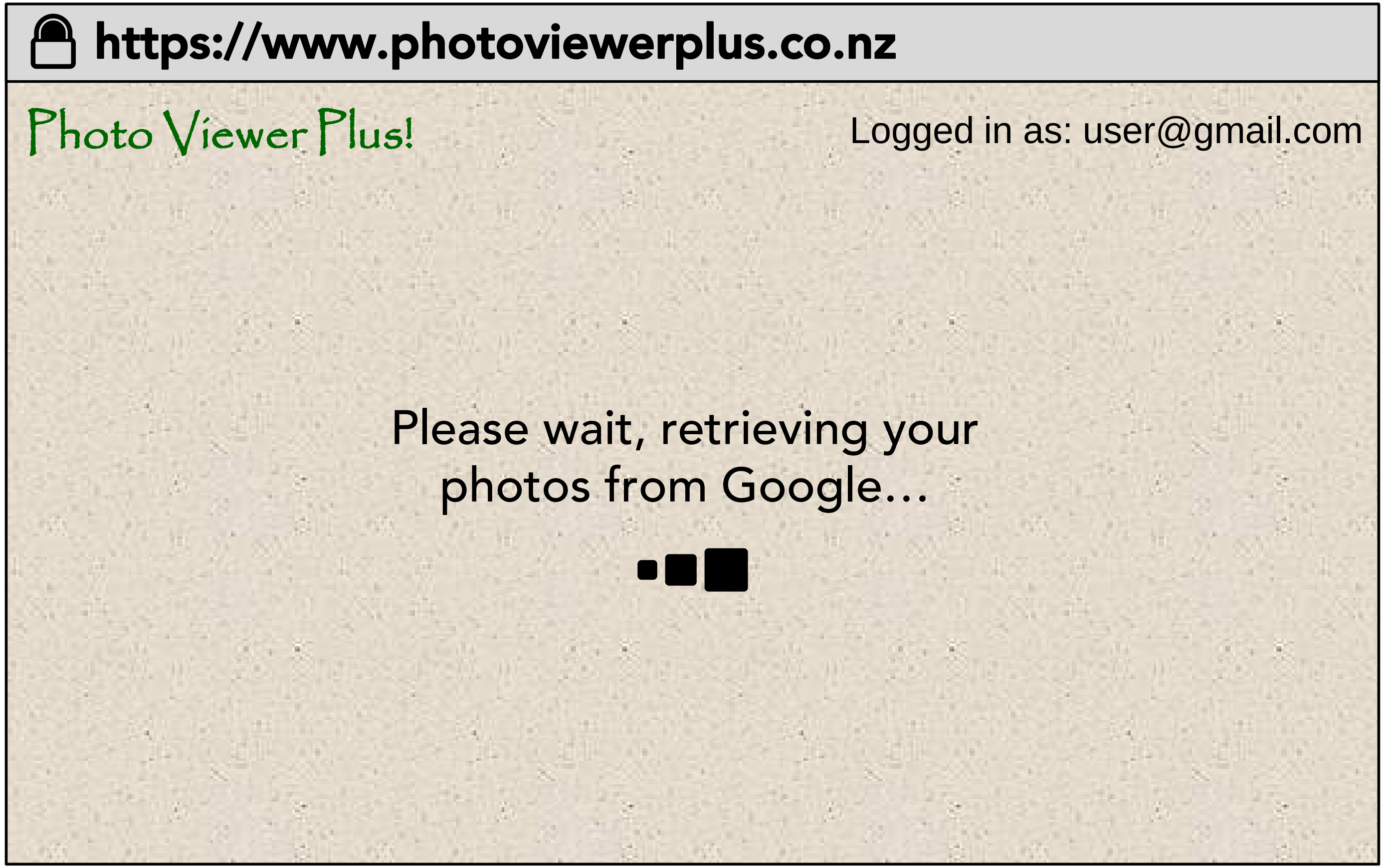
user@gmail.com

This will allow Sample Application to:

View, modify and delete your photos

Allow Sample Application to do this?

You may review this app's privacy policies. You can view or remove app access in your Google Account



https://www.photoviewerplus.co.nz

Photo Viewer Plus!

Logged in as: user@gmail.com

Please wait, retrieving your photos from Google…

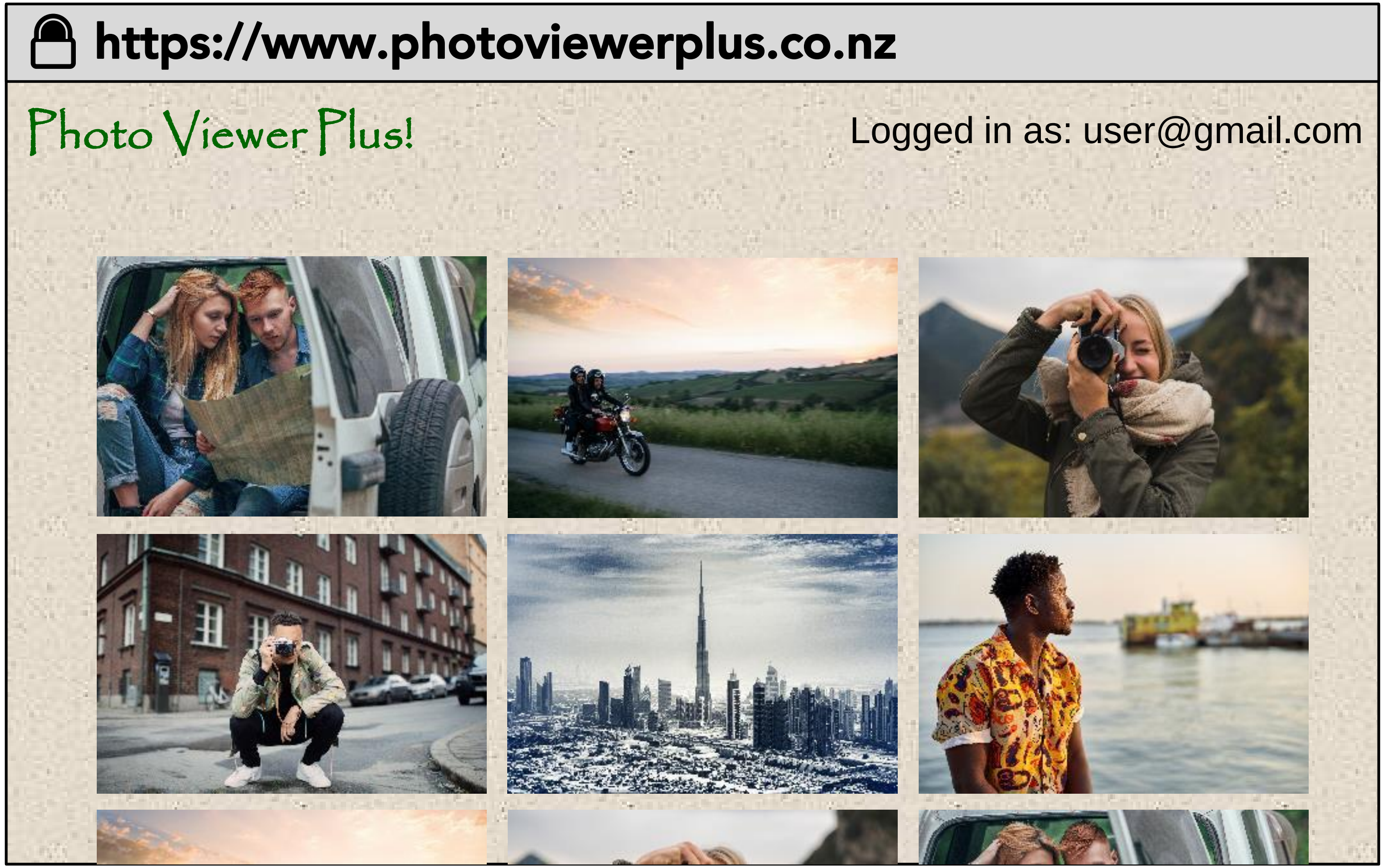
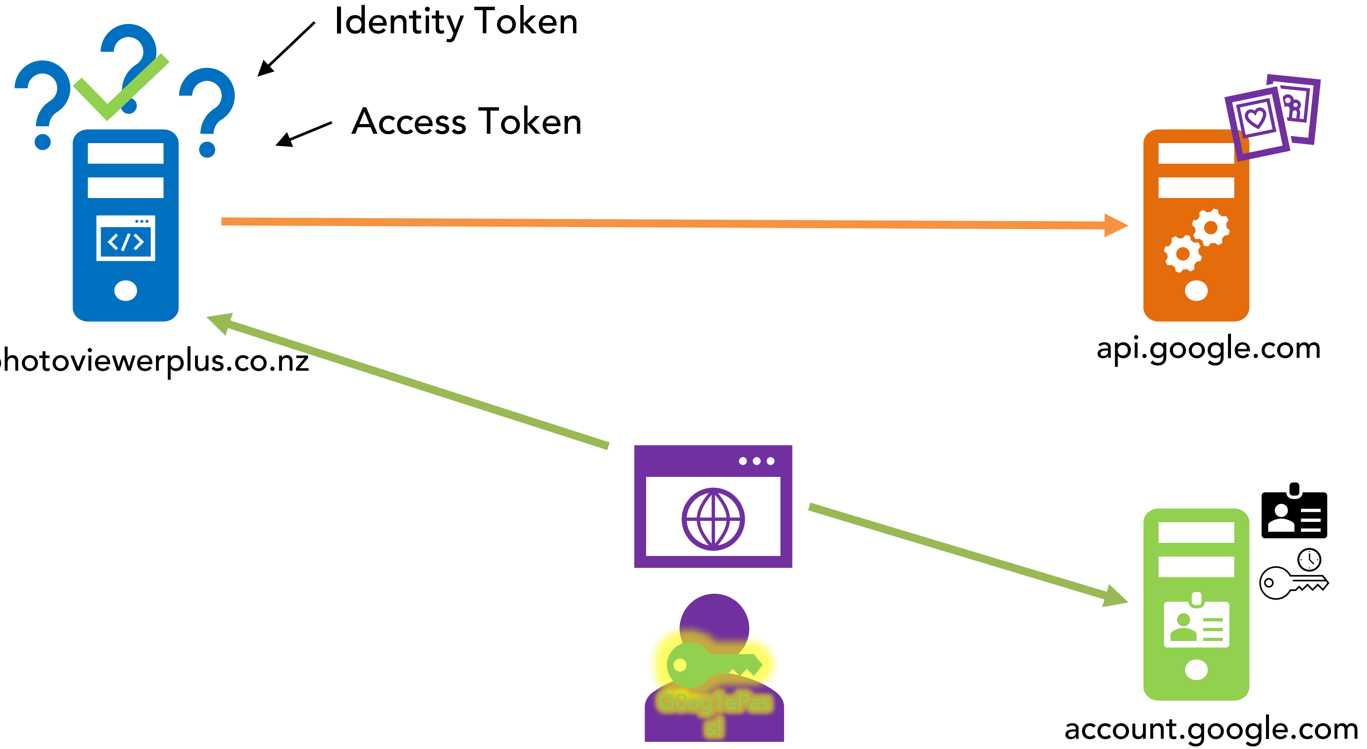

https://www.photoviewerplus.co.nz

Photo Viewer Plus!

Logged in as: user@gmail.com

What just happened?



Identity Token

Tells photoviewerplus.co.nz who Matt Cotterell is:

- Some info about them
- How they logged on
- Groups they belong to

Access Token

Lets photoviewerplus.co.nz *briefly* access Matt Cotterell's data on api.photos.google.com:

- Photos (Read + Write)
- Favourites List (Read Only)

There's *three* servers now?

Let’s break it down:

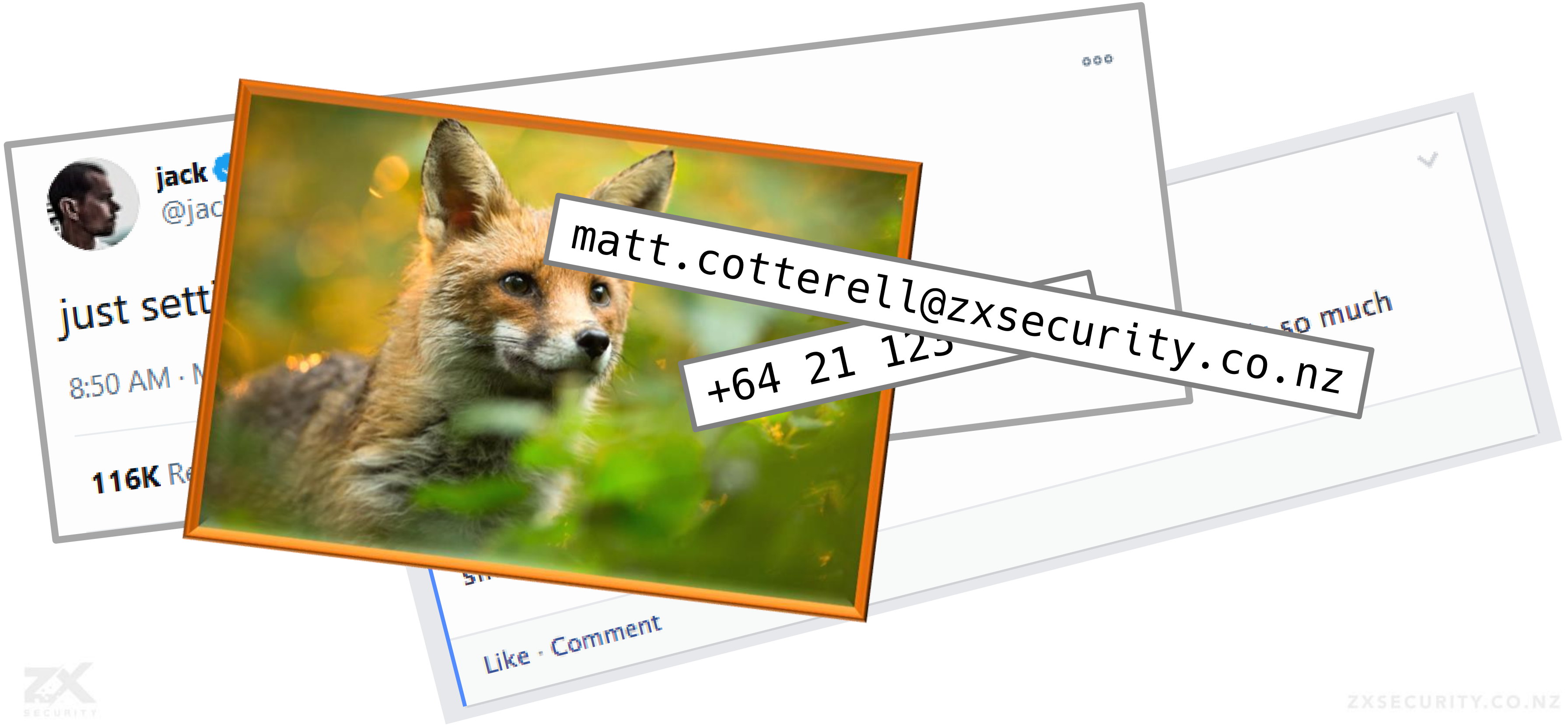
Resource

“Stuff”

Terminology

Resource

"Stuff"



Let’s break it down:

Resource
“Stuff”

Resource Owner
“User”

Terminology

Resource Owner

"User"

Let’s break it down:

Resource
“Stuff”

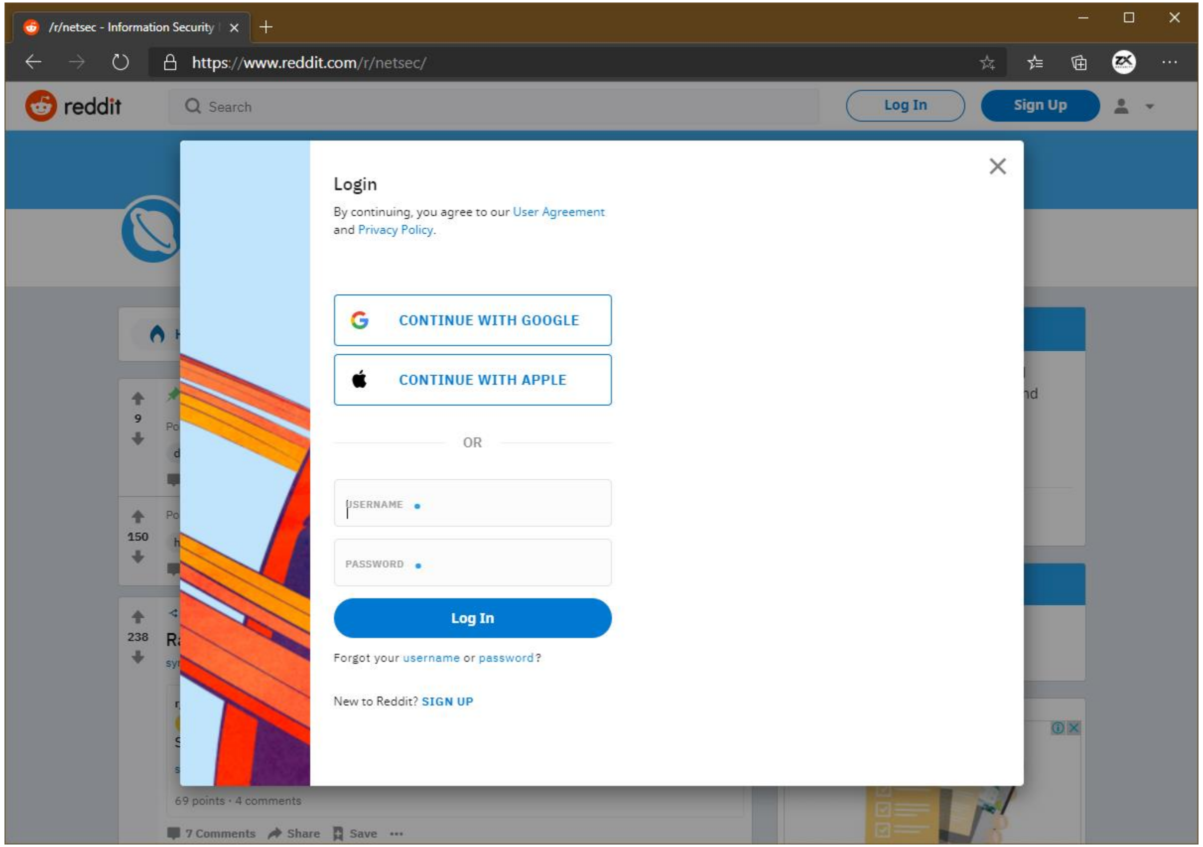
Resource Owner
“User”

Client
“App” / “Server”

Terminology

Client

"App" / "Server"



/r/netsec - Information Security

https://www.reddit.com/r/netsec/

reddit

Search

Log In

Sign Up

Login

By continuing, you agree to our User Agreement and Privacy Policy.

CONTINUE WITH GOOGLE

CONTINUE WITH APPLE

OR

USERNAME

PASSWORD

Log In

Forgot your username or password?

New to Reddit? SIGN UP

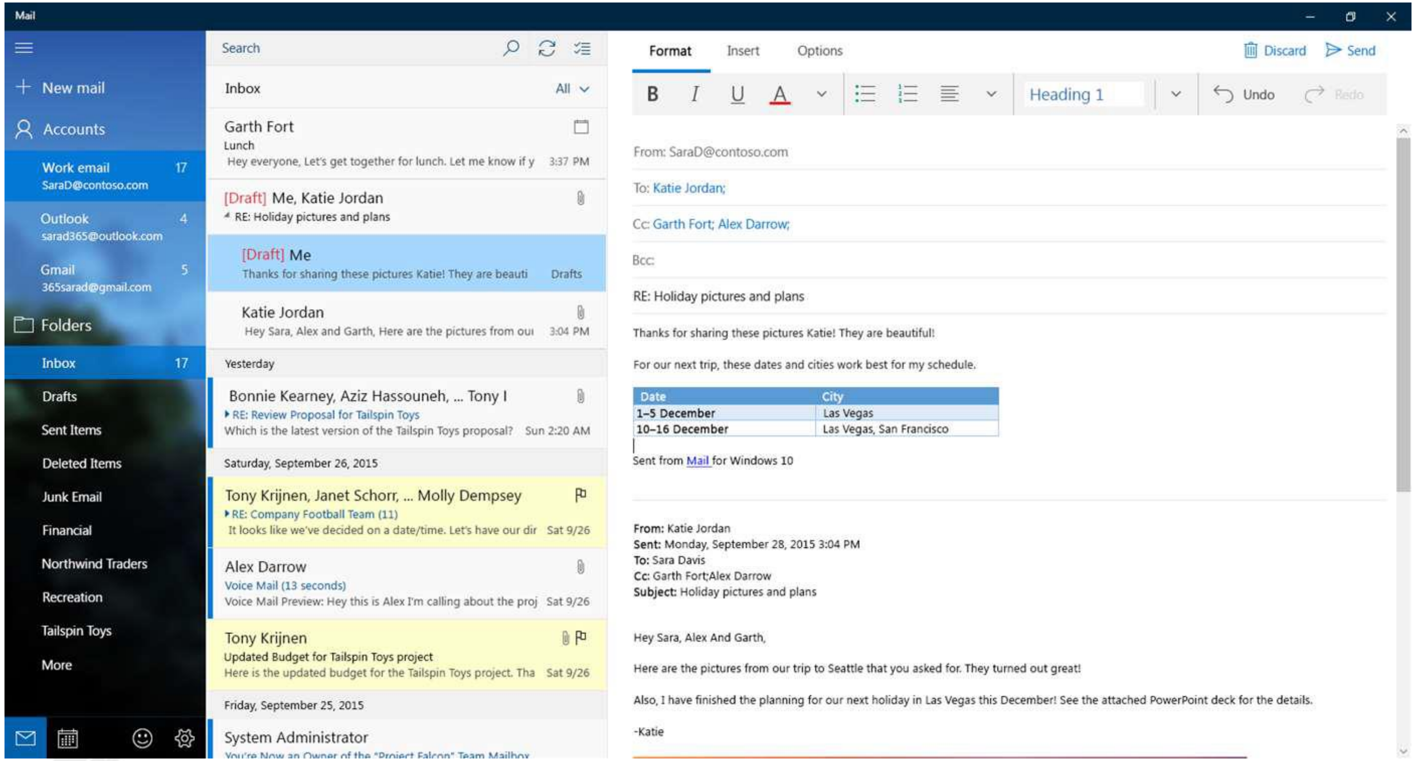
69 points · 4 comments

7 Comments Share Save

Terminology

Client

"App" / "Server"



Mail

Search

New mail

Accounts

Work email 17
SaraD@contoso.com

Outlook 4
sarad365@outlook.com

Gmail 5
365sarad@gmail.com

Folders

Inbox 17
Drafts
Sent Items
Deleted Items
Junk Email
Financial
Northwind Traders
Recreation
Tailspin Toys
More

Inbox All

Garth Fort
Lunch
Hey everyone, Let's get together for lunch. Let me know if y 3:37 PM

[Draft] Me, Katie Jordan
RE: Holiday pictures and plans

[Draft] Me
Thanks for sharing these pictures Katie! They are beauti Drafts

Katie Jordan
Hey Sara, Alex and Garth, Here are the pictures from our 3:04 PM

Yesterday

Bonnie Kearney, Aziz Hassouneh, ... Tony I
RE: Review Proposal for Tailspin Toys
Which is the latest version of the Tailspin Toys proposal? Sun 2:20 AM

Saturday, September 26, 2015

Tony Krijnen, Janet Schorr, ... Molly Dempsey
RE: Company Football Team (11)
It looks like we've decided on a date/time. Let's have our dir Sat 9/26

Alex Darrow
Voice Mail (13 seconds)
Voice Mail Preview: Hey this is Alex I'm calling about the proj Sat 9/26

Tony Krijnen
Updated Budget for Tailspin Toys project
Here is the updated budget for the Tailspin Toys project. Tha Sat 9/26

Friday, September 25, 2015

System Administrator
You're Now an Owner of the "Project Falcon" Team Mailbox

Format Insert Options Discard Send

B I U A Heading 1 Undo Redo

From: SaraD@contoso.com

To: Katie Jordan;

Cc: Garth Fort; Alex Darrow;

Bcc:

RE: Holiday pictures and plans

Thanks for sharing these pictures Katie! They are beautiful!

For our next trip, these dates and cities work best for my schedule.

Date	City
1–5 December	Las Vegas
10–16 December	Las Vegas, San Francisco

Sent from Mail for Windows 10

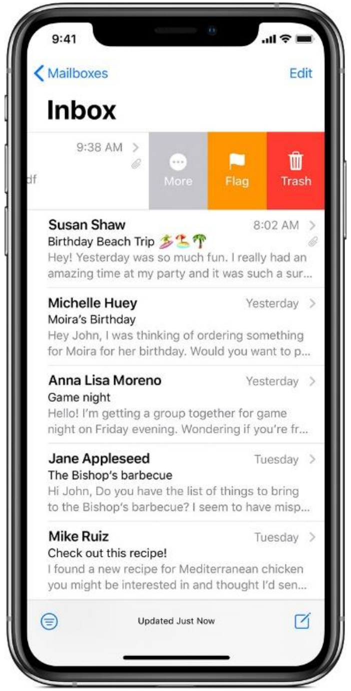
From: Katie Jordan
Sent: Monday, September 28, 2015 3:04 PM
To: Sara Davis
Cc: Garth Fort;Alex Darrow
Subject: Holiday pictures and plans

Hey Sara, Alex And Garth,

Here are the pictures from our trip to Seattle that you asked for. They turned out great!

Also, I have finished the planning for our next holiday in Las Vegas this December! See the attached PowerPoint deck for the details.

-Katie



9:41

Mailboxes Edit

Inbox

9:38 AM
More Flag Trash

Susan Shaw 8:02 AM
Birthday Beach Trip
Hey! Yesterday was so much fun. I really had an amazing time at my party and it was such a sur...

Michelle Huey Yesterday
Moira's Birthday
Hey John, I was thinking of ordering something for Moira for her birthday. Would you want to p...

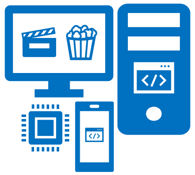
Anna Lisa Moreno Yesterday
Game night
Hello! I'm getting a group together for game night on Friday evening. Wondering if you're fr...

Jane Appleseed Tuesday
The Bishop's barbecue
Hi John, Do you have the list of things to bring to the Bishop's barbecue? I seem to have misp...

Mike Ruiz Tuesday
Check out this recipe!
I found a new recipe for Mediterranean chicken you might be interested in and thought I'd sen...

Updated Just Now

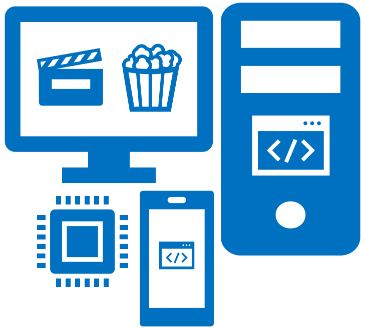
Terminology



Client

"App" / "Server"

Terminology



"The thing with the 'log in' button."

Let’s break it down:

Resource
“Stuff”

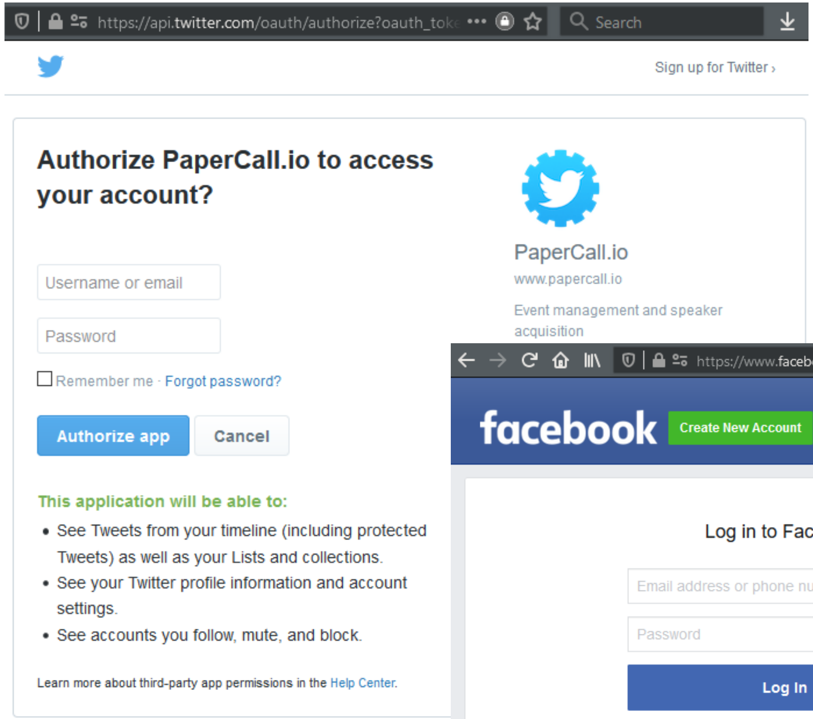
Resource Owner
“User”

Client
“App” / “Server”

Identity Provider
“Authorisation Server”

Terminology

Authorization Server
"Identity Provider"


https://api.twitter.com/oauth/authorize?oauth_toke

Sign up for Twitter ›

Authorize PaperCall.io to access your account?

Username or email

Password

Remember me · Forgot password?

Authorize app | Cancel

PaperCall.io

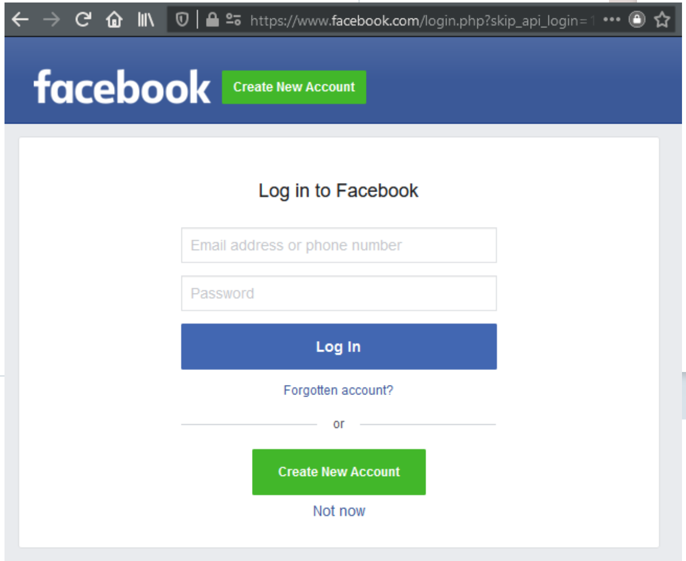
www.papercall.io

Event management and speaker acquisition

This application will be able to:

- See Tweets from your timeline (including protected Tweets) as well as your Lists and collections.
- See your Twitter profile information and account settings.
- See accounts you follow, mute, and block.

Learn more about third-party app permissions in the Help Center.


https://www.facebook.com/login.php?skip_api_login=

facebook | Create New Account

Log in to Facebook

Email address or phone number

Password

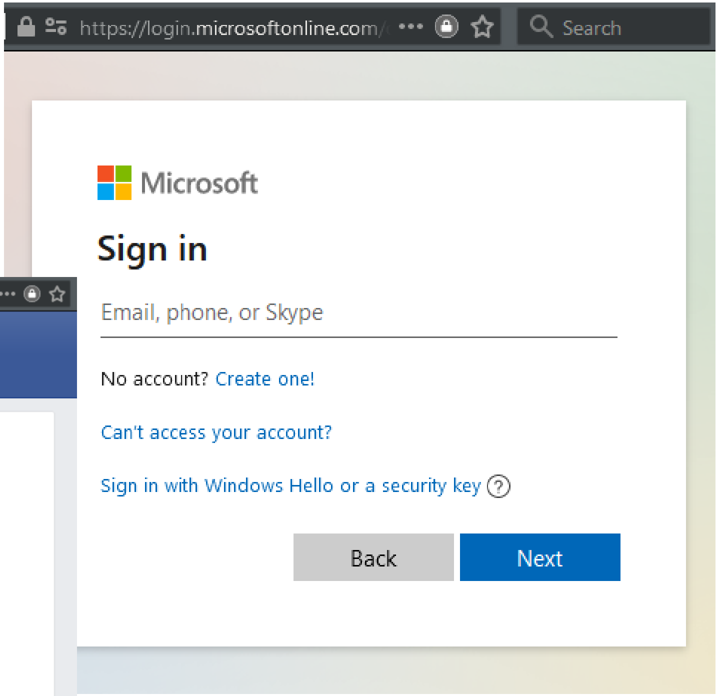
Log In

Forgotten account?

or

Create New Account

Not now


https://login.microsoftonline.com/

Microsoft

Sign in

Email, phone, or Skype

No account? Create one!

Can't access your account?

Sign in with Windows Hello or a security key

Back | Next

Terminology

"The place where you prove who you are"

Let’s break it down:

Resource
“Stuff”

Resource Owner
“User”

Client
“App” / “Server”

Resource Server
“API Server”

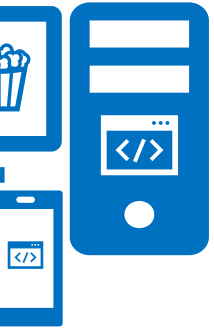
Identity Provider
“Authorisation Server”

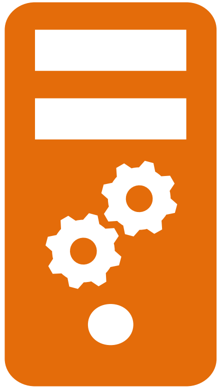
Terminology

Resource Server
"API Server"

api.photos.google.com

Terminology





"Where the stuff is that the Client wants."

Tokens?

ZXSECURITY.CO.NZ

Two (general) ways tokens can work…

“Reference” Tokens

- Usually just a random string
- Often references some other data (like in a database)
- Database: “`xXBw2AAAABlBMVEX` is a session ID for User 159”

“Self-contained” Tokens

- Has data inside it
- Uses cryptography to make sure it’s not fake or altered
- Can be verified “offline”
- Examples: JWT, PASETO, SAML Assertions…

JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMj
M0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxN
TE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6
yJV_adQssw5c

JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMj
M0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxN
TE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6
yJV_adQssw5c

JWT

eyJhbGciOiJIUzI1NiIsInR5c
CI6IkpXVCJ9

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

eyJzdWIiOiIxMjM0NTY3O
DkwIiwibmFtZSI6IkpvaG4g
RG9lIiwiaWF0IjoxNTE2MjM
5MDIyfQ

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

HMAC256(header + payload, secret)

SflKxwRJSMeKKF2QT4fwp
MeJf36POk6yJV_adQssw5

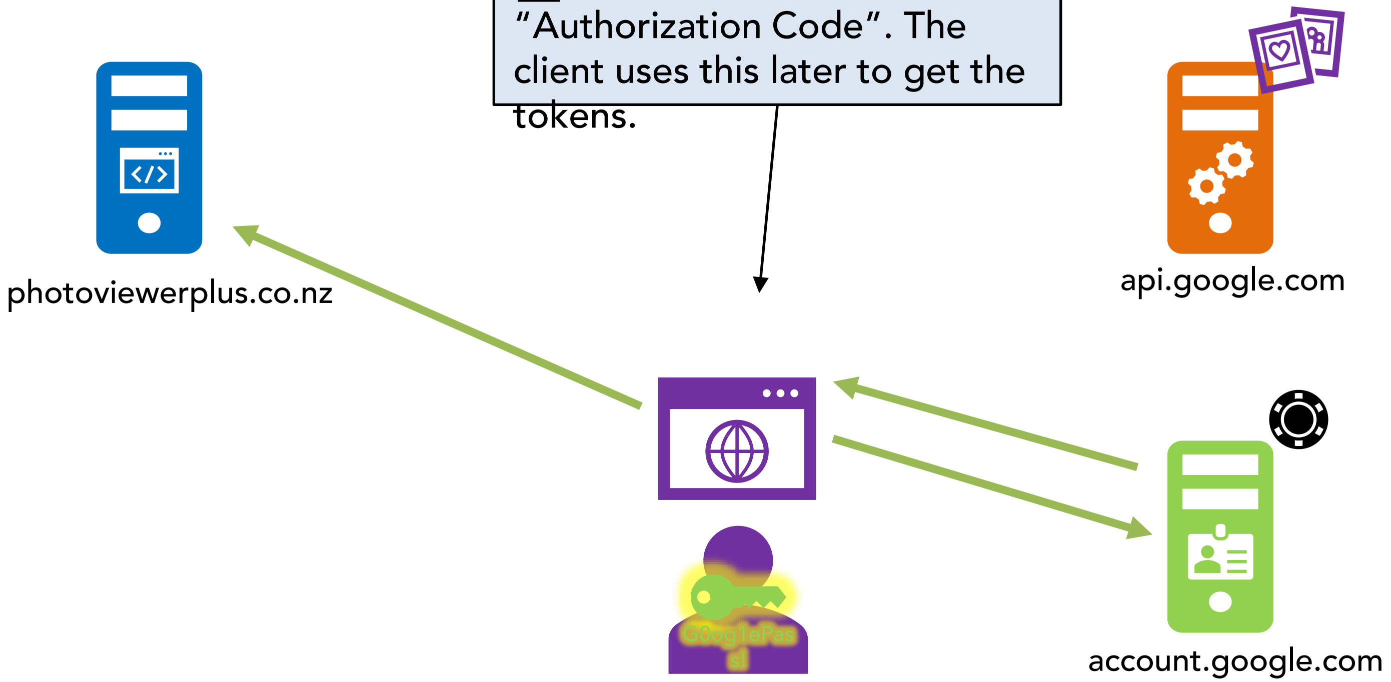
How does it work?

OAuth 2.0 Flows – Authorisation Code

- Most common flow you'll see in web applications
- Two steps:
    1. Resource Owner logs in to Identity Provider using browser, gets code
    2. Client exchanges code with Identity Provider for access/id tokens

OAuth 2.0 Flows – Authorisation Code (Step 1)

i This is called the "Authorization Code". The client uses this later to get the tokens.



OAuth 2.0 Flows – Authorisation Code (Step 1)

```
https://account.google.com/authorise
?response_type=code
&client_id=1234567890-abc123def456
&redirect_uri=https://photoviewerplus.co.nz/callback
&scope=photo.read+premium.promote_photo
&state=xcoiv98y2kd22vusuye3kch
```

response_type=code: Use the Authorization Code flow

client_id: Public identifier for this client (photoviewerplus.co.nz)

redirect_uri: Go back here when user is finished logging in

scope: Limits what the client is allowed to access ("photo.read" and "premium.promote_photo")

state: Similar to a "CSRF" token. The client picks a non-guessable value. The Authorization Server has to send the same value back.

OAuth 2.0 Flows – Authorisation Code (Step 1)

```
HTTP/1.1 200 OK
Location: https://photoviewerplus.co.nz/callback
 ?code=g0ZGZmNjVmOWIjNTk2NTk4ZTYyZGI3
 &state=xcoiv98y2kd22vusuye3kch
```

i The resource owner gives this to the client. The client exchanges it for the access/id token!

OAuth 2.0 Flows – Authorisation Code (Step 1)

```
https://account.google.com/authorise
?response_type=code
&client_id=1234567890-abc123def456
&redirect_uri=https://photoviewerplus.co.nz/callback
&scope=photo.read+premium.promote_photo
&state=xcoiv98y2kd22vusuye3kch
```

response_type=code: Use the Authorization Code flow

client_id: Public identifier for this **client** (photoviewerplus.co.nz)

redirect_uri: Go back here when user is finished logging in

scope: Limits what the **client** is allowed to access ("photo.read" and "premium.promote_photo")

state: Similar to a "CSRF" token. The **client** picks a non-guessable value. The **Identity Provider** has to send the same value back.

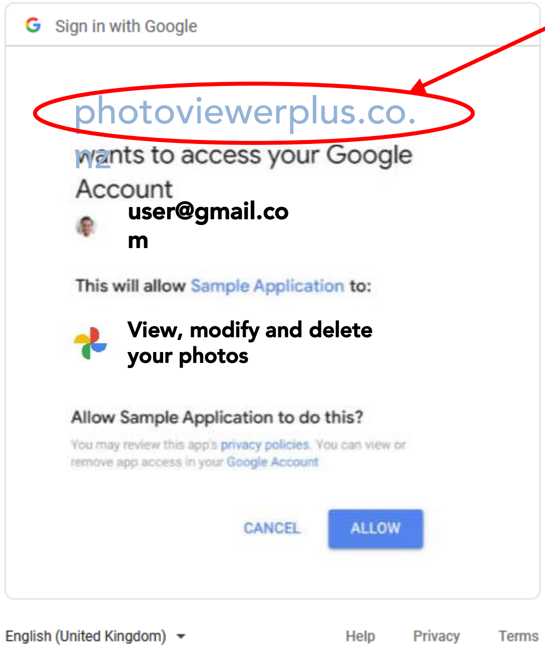
Important note about Redirect URLs!

```
https://account.google.com/authorise
?response_type=code
&client_id=1234567890-abc123def456
&redirect_uri=https://evilbadsite.com/hack
&scope=photo.read+premium.promote_photo
&state=xcoiv98y2kd22vusuye3kch
```

Important note about Redirect URLs!

⚠ Remember, Client ID is public, anyone can find it!

https://account.google.com/authorize?client_id=1234567890[…]



Sign in with Google

photoviewerplus.co.nz wants to access your Google Account

user@gmail.com

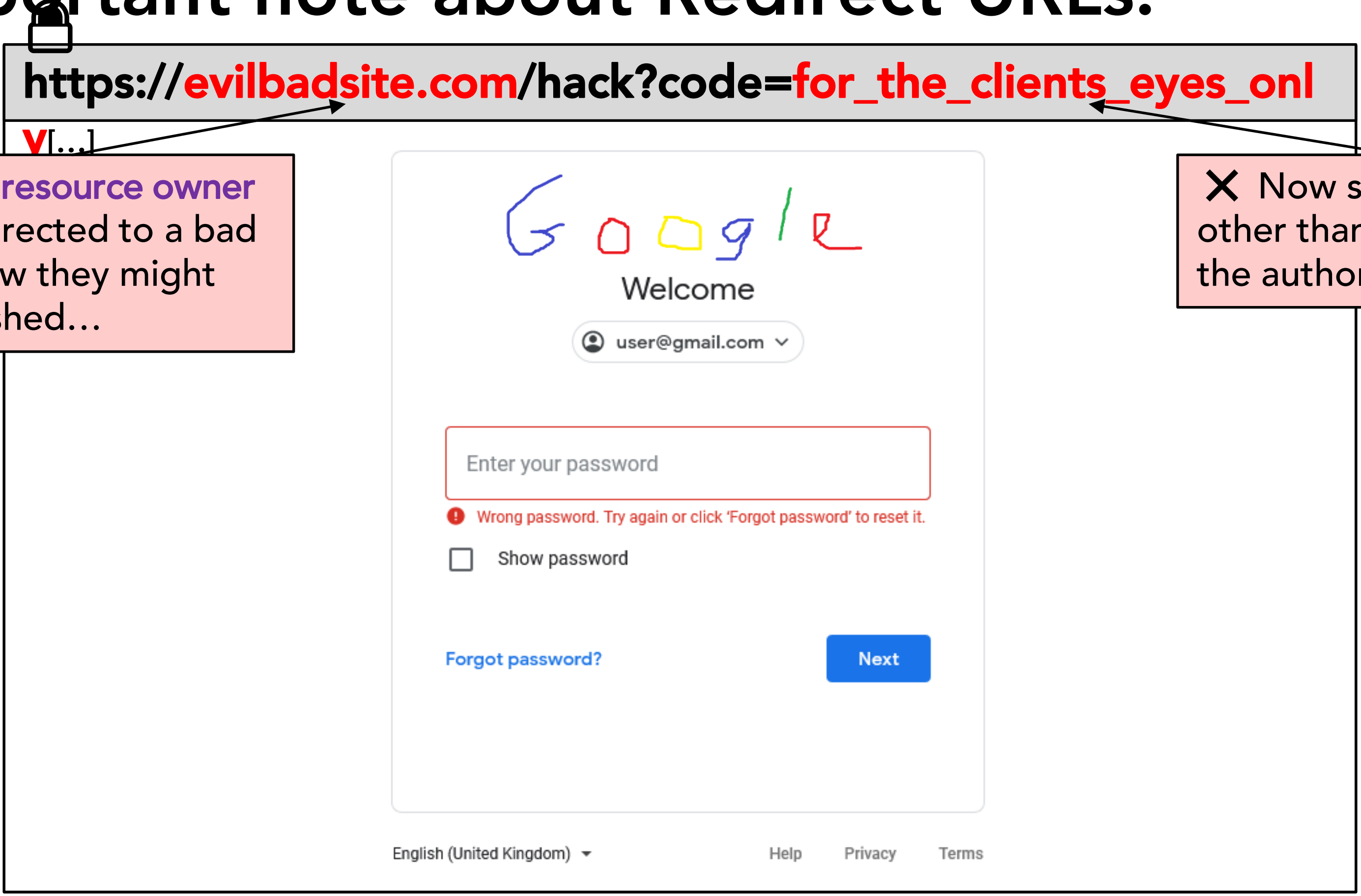
This will allow Sample Application to:

View, modify and delete your photos

Allow Sample Application to do this?

You may review this app's privacy policies. You can view or remove app access in your Google Account

Important note about Redirect URLs!


https://evilbadsite.com/hack?code=for_the_clients_eyes_onl
y[...]

✕ The resource owner got redirected to a bad site! Now they might get phished…

✕ Now somebody other than the client got the authorisation code!

Important note about Redirect URLs!

- Extremely important this only goes to expected URLs!
    - Add redirect URLs to a safelist
    - Client ID should ONLY allow return to certain URLs!

OAuth 2.0 Flows – Authorisation Code (Step 1)

```
https://account.google.com/authorise
?response_type=code
&client_id=1234567890-abc123def456
&redirect_uri=https://photoviewerplus.co.nz/callback
&scope=photo.read+premium.promote_photo
&state=xcoiv98y2kd22vusuye3kch
```

response_type=code: Use the Authorization Code flow

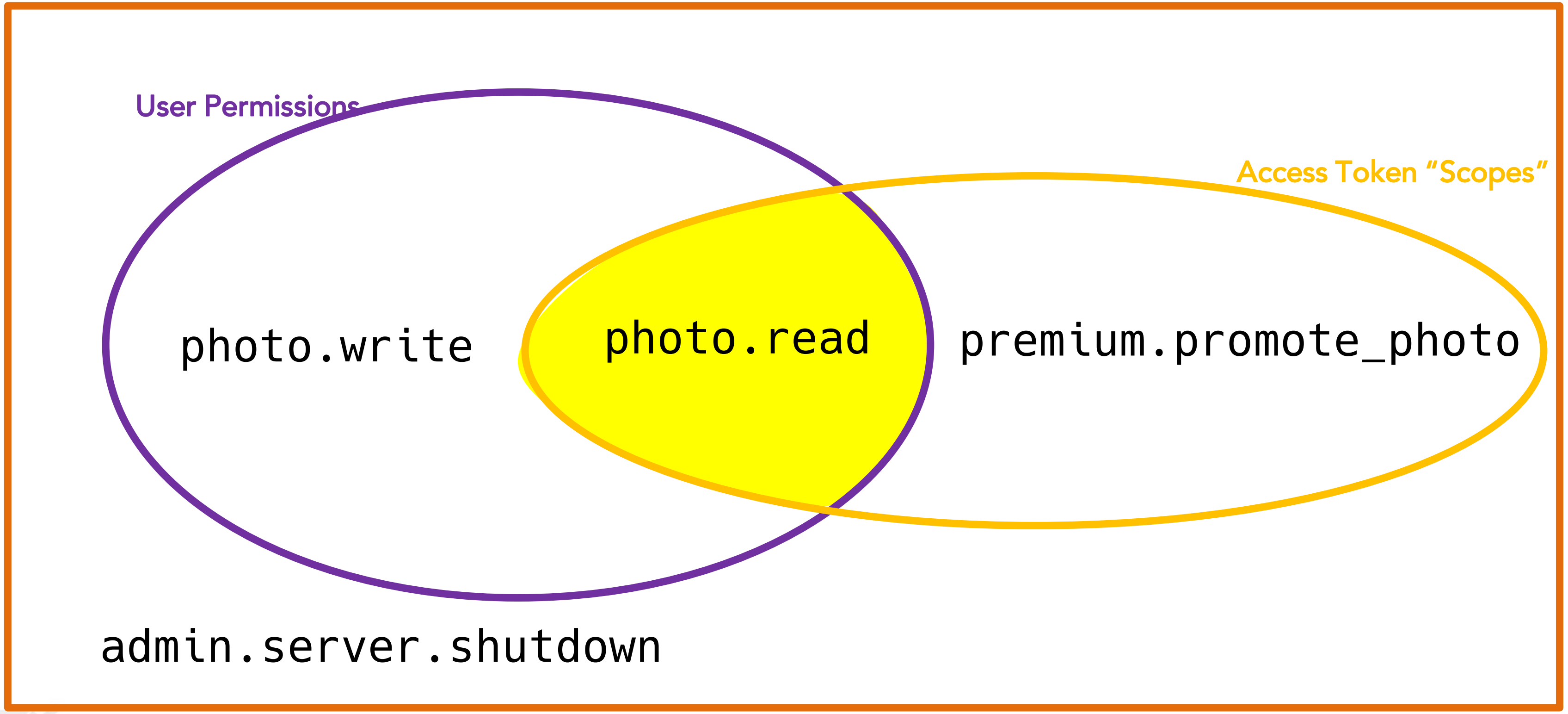
client_id: Public identifier for this **client** (photoviewerplus.co.nz)

redirect_uri: Go back here when user is finished logging in

scope: Limits what the **client** is allowed to access ("photo.read" and "premium.promote_photo")

state: Similar to a "CSRF" token. The **client** picks a non-guessable value. The Identity Provider has to send the same value back.

OAuth 2.0 Scopes



api.google.com

OAuth 2.0 Scopes

- Scopes limit what the access token can do
  - **Client** can't do more than the **resource owner** can do
  - **Resource Owner** can limit what the **client** can do
- Scopes can also limit things in the identity token
  - **Resource Owner** decides what personal details the **client** gets

OAuth 2.0 Flows – Authorisation Code (Step 1)

```
https://account.google.com/authorise
?response_type=code
&client_id=1234567890-abc123def456
&redirect_uri=https://photoviewerplus.co.nz/callback
&scope=photo.read+premium.promote_photo
&state=xcoiv98y2kd22vusuye3kch
```

response_type=code: Use the Authorization Code flow

client_id: Public identifier for this client (photoviewerplus.co.nz)

redirect_uri: Go back here when user is finished logging in

scope: Limits what the client is allowed to access ("photo.read" and "premium.promote_photo")

state: Similar to a "CSRF" token. The client picks a non-guessable value. The Identity Provider has to send the same value back.

The "state" parameter

Think of it like a CSRF token…

1. Client decides on any (non-guessable) value
2. Identity Provider simply returns it with the authorisation code
3. Client checks:
   - Is this a "state" value I created?
   - Is this the same browser I gave it to? (check cookies, etc)

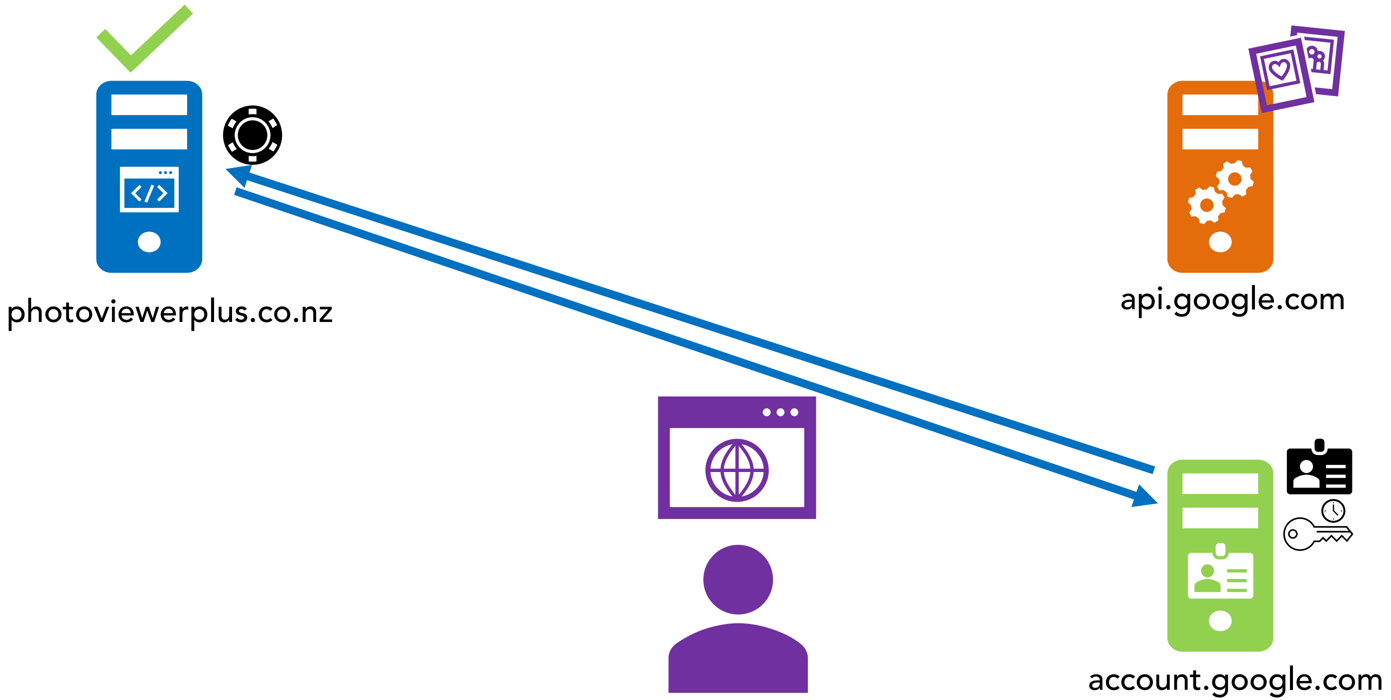
The “state” parameter

Without it:

- Victim clicks link with attackers own authorisation code
- Victim is logged into attackers account
- Victim enters personal information into attackers account

Search for **“Session Fixation Attack”** for more information!

OAuth 2.0 Flows – Authorisation Code (Step 2)



OAuth 2.0 Flows – Authorisation Code (Step 2)

```
POST /token HTTP/1.1
Host: account.google.com

grant_type=authorization_code
&code=g0ZGZmNjVmOWIjNTk2NTk4ZTYyZGI3
&redirect_uri=https://photoviewerplus.co.nz/callback
&client_id=1234567890-abc123def456
&client_secret=qWgdYAmab0YSkuL1qKv5bPX
```

response_type=authorization_code: Redeem an authorization code for access/id tokens

code: The Authorization code we got from the previous step

redirect_uri: Where the resource owner was sent to when they finished logging in

client_id: Public identifier for this client (photoviewerplus.co.nz)

client_secret: "Password" that only the client knows

OAuth 2.0 Flows – Authorisation Code (Step 2)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token":"eyJ0NjJkZmQ5OTM2[…]NDE1ZTZjNGZmZjI3",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"eYJGYzYTlmM2YxO[…]TQ5MGE3YmNmMDFkNTVk",
  "scope":"photo.read",
  "id_token": "eyJzcifQewoNz[…]AKfQggW8hMzqg"
}
```

i Access tokens don't last very long! The **client** can use this "Refresh Token" to get a fresh one without bothering the **user**.

What about things that aren't websites?

Lots of different scenarios available

- What if there is no **user**? (service-to-service)
- What if the **user** doesn't have a keyboard? (IoT, Smart TV, etc)

OAuth 2.0 Flows

Authorisation Code

Resource Owner wants to give Client access to Resource Server

Mobile Apps, Web, Desktop Clients

Client Credentials

Client wants to access Resource Server

No Resource Owner Involved

"Server to Server"

Device Code

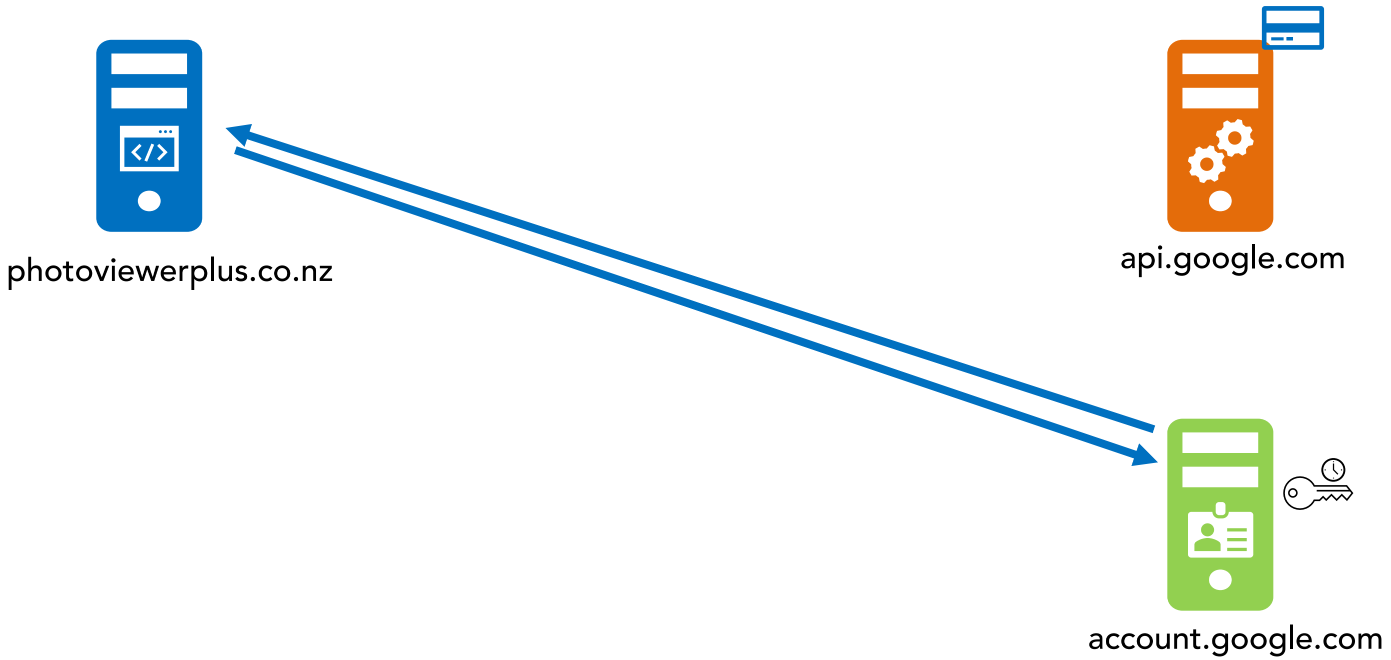
Resource Owner can't easily auth with a web view

IoT, Smart TVs, CLI…

~~**Implicit Flow**~~

~~**Resource Owner Password Credential**~~

OAuth 2.0 Flows – Client Credentials



OAuth 2.0 Flows – Client Credenti

This is very similar to "Step 2" before! The difference is there is no Authorization Code, because there is no **user**.

```
POST /token HTTP/1.1
Host: accounts.google.com
Content-Type: application/x-www-form-
urlencoded

grant_type=client_credentials
&client_id=1234567890-abc123def456
&client_secret=qWgdYAmab0YSkuL1qKv5bPX
&scope=billing+licenses
```

response_type=client_credentials: Use the Client Credentials flow

client_id: Public identifier for this **client** (photoviewerplus.co.nz)

client_secret: "Password" that only the **client** knows

scope: Limits what the **client** can do with the token it gets ("billing" and "licenses")

OAuth 2.0 Flows – Client Credentials

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token":“eyJ0NjJkZmQ5OTM2[…]NDE1ZTZjNGZmZjI3",
  "token_type":"bearer",
  "expires_in":3600,
  "scope":“billing“
}
```

Lots of different scenarios available

- What if there is no user? (service-to-service)
- What if the user doesn't have a keyboard? (IoT, Smart TV, etc)

OAuth 2.0 Flows

Authorisation Code

Resource Owner wants to give Client access to Resource Server

Mobile Apps, Web, Desktop Clients

Client Credentials

Client wants to access Resource Server

No Resource Owner Involved

"Server to Server"

Device Code

Resource Owner can't easily auth with a web view

IoT, Smart TVs, CLI…

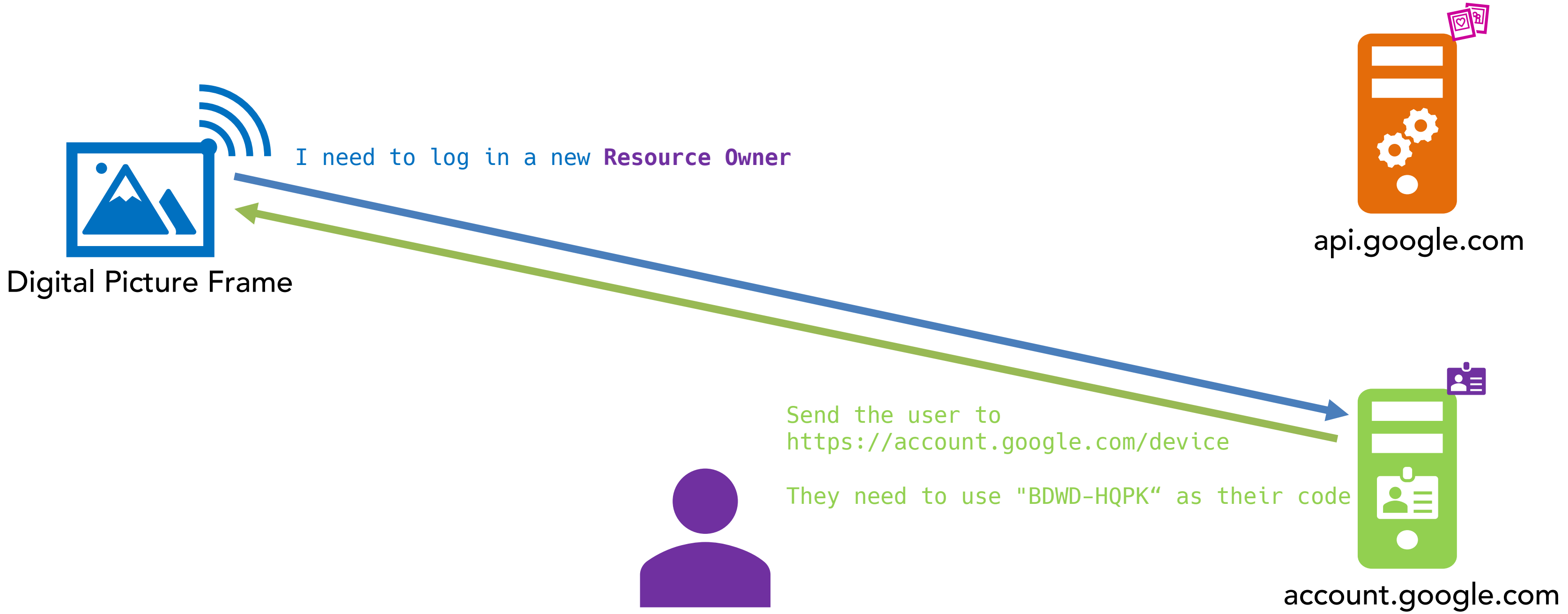
Implicit Flow

Resource Owner Password Credential

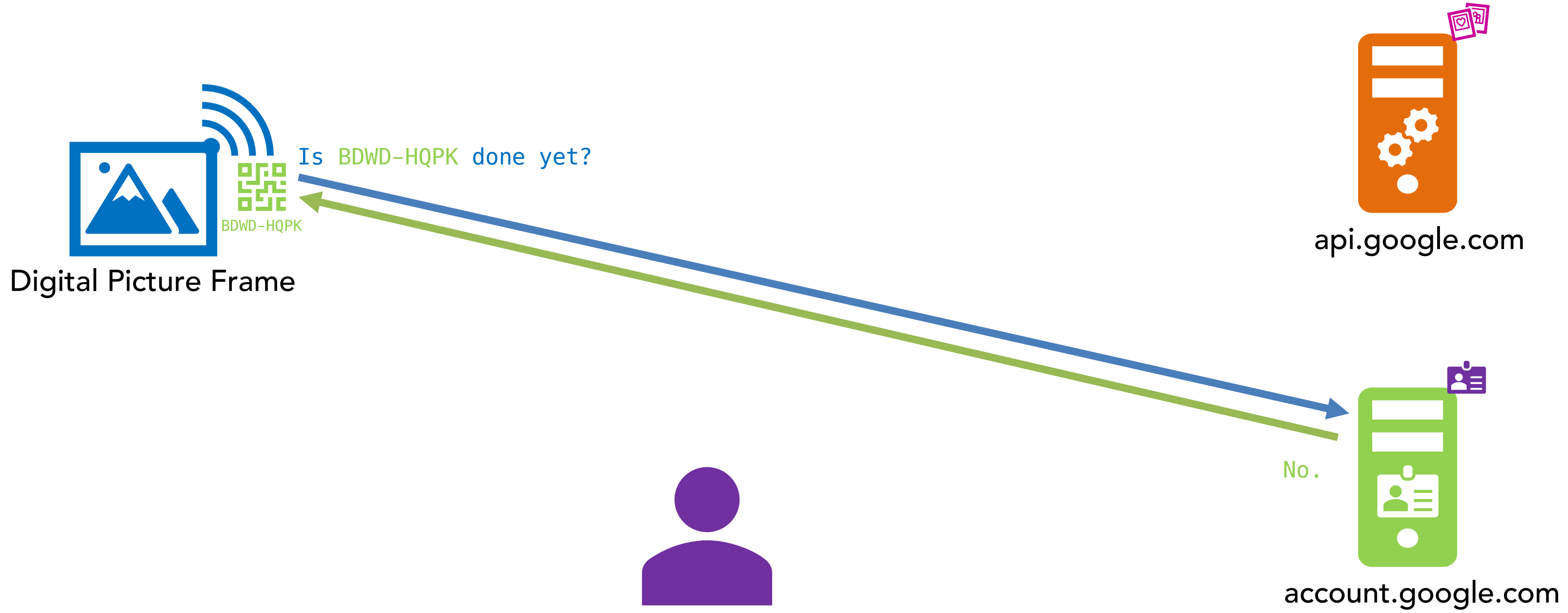
OAuth 2.0 Flows – Device Code

Scan
this:
Use this code when asked:
ABCD123
4
kogan

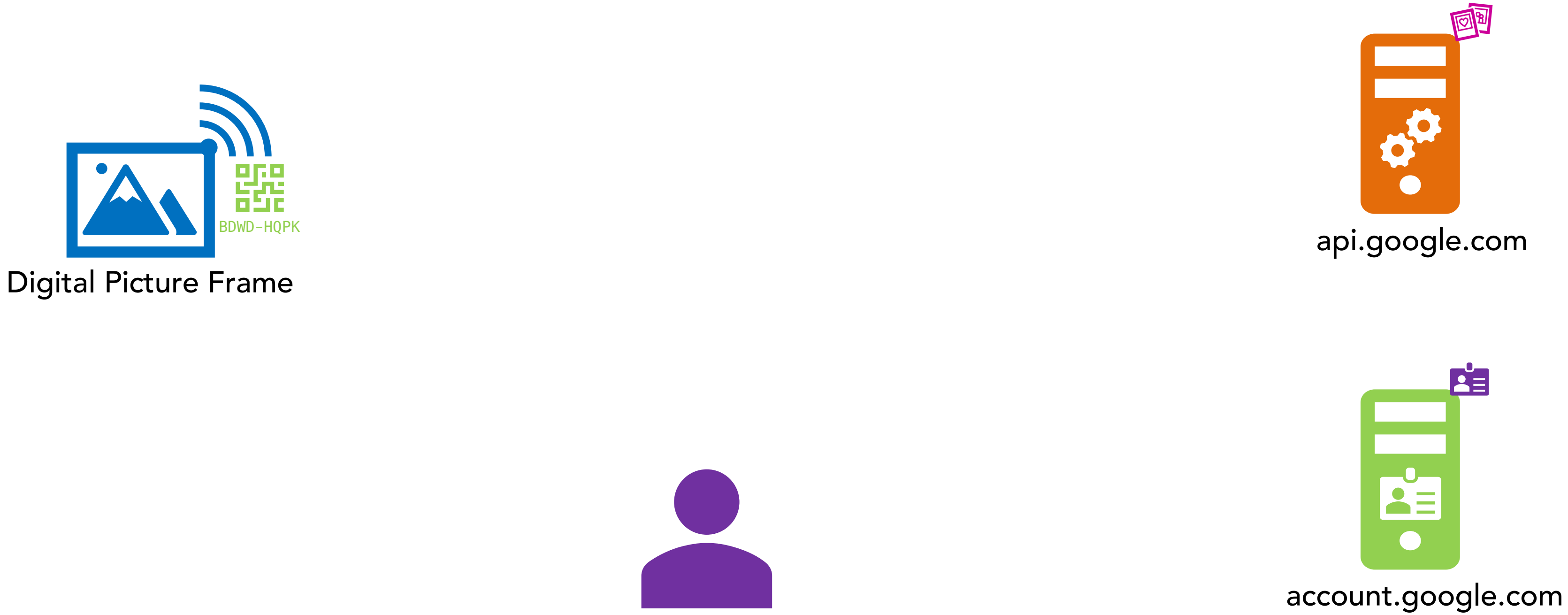
OAuth 2.0 Flows – Device Code


I need to log in a new **Resource Owner**
Digital Picture Frame
api.google.com
Send the user to
https://account.google.com/device
They need to use "BDWD-HQPK“ as their code
account.google.com

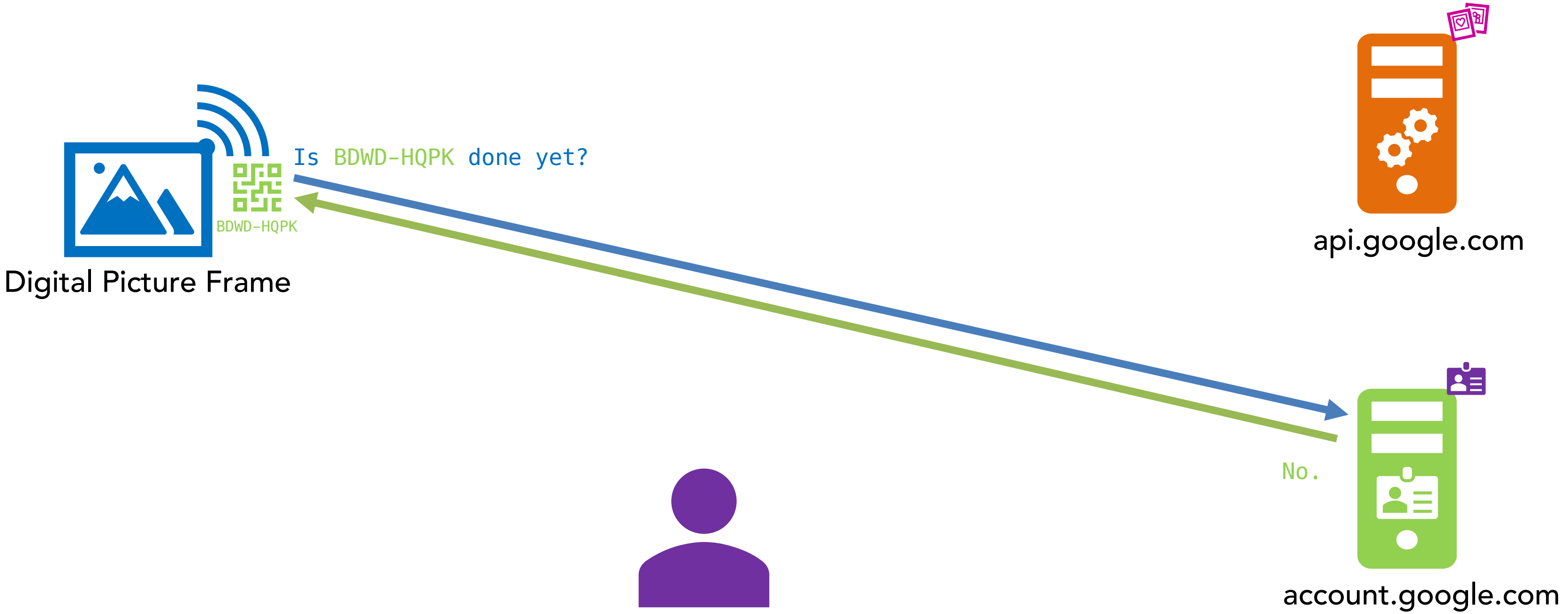
OAuth 2.0 Flows – Device Code



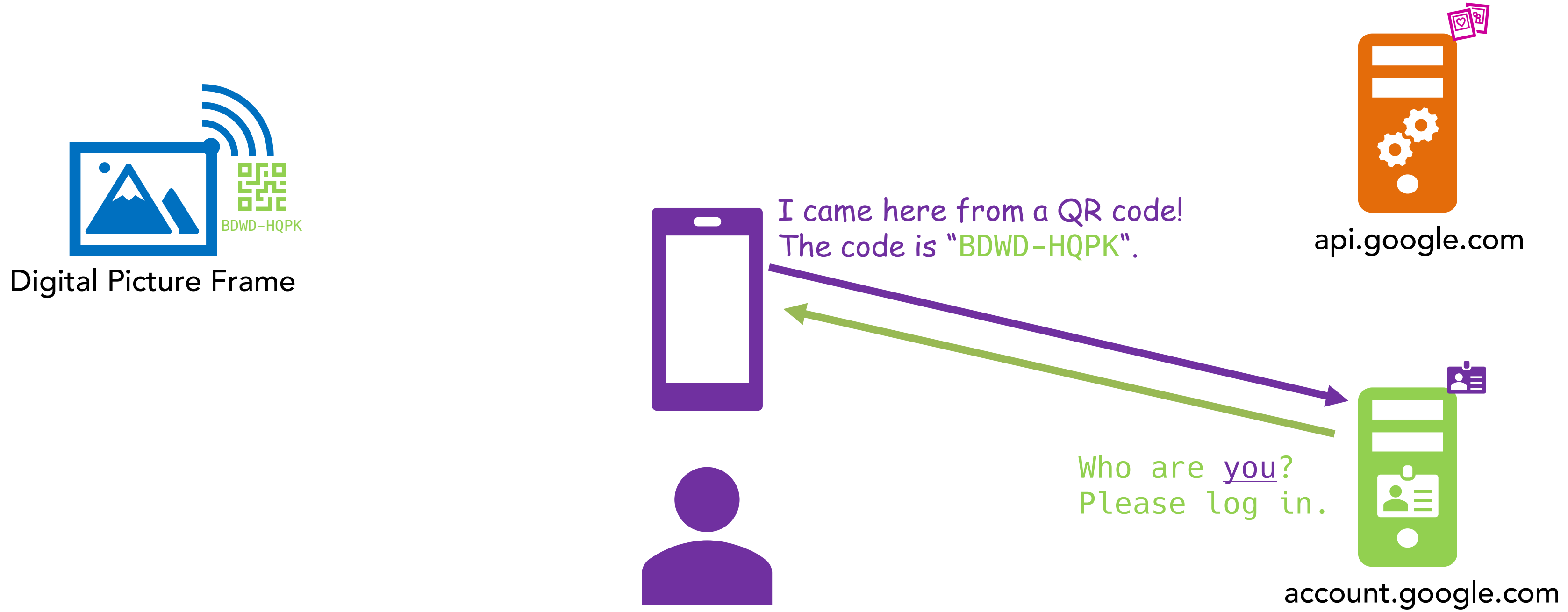
OAuth 2.0 Flows – Device Code



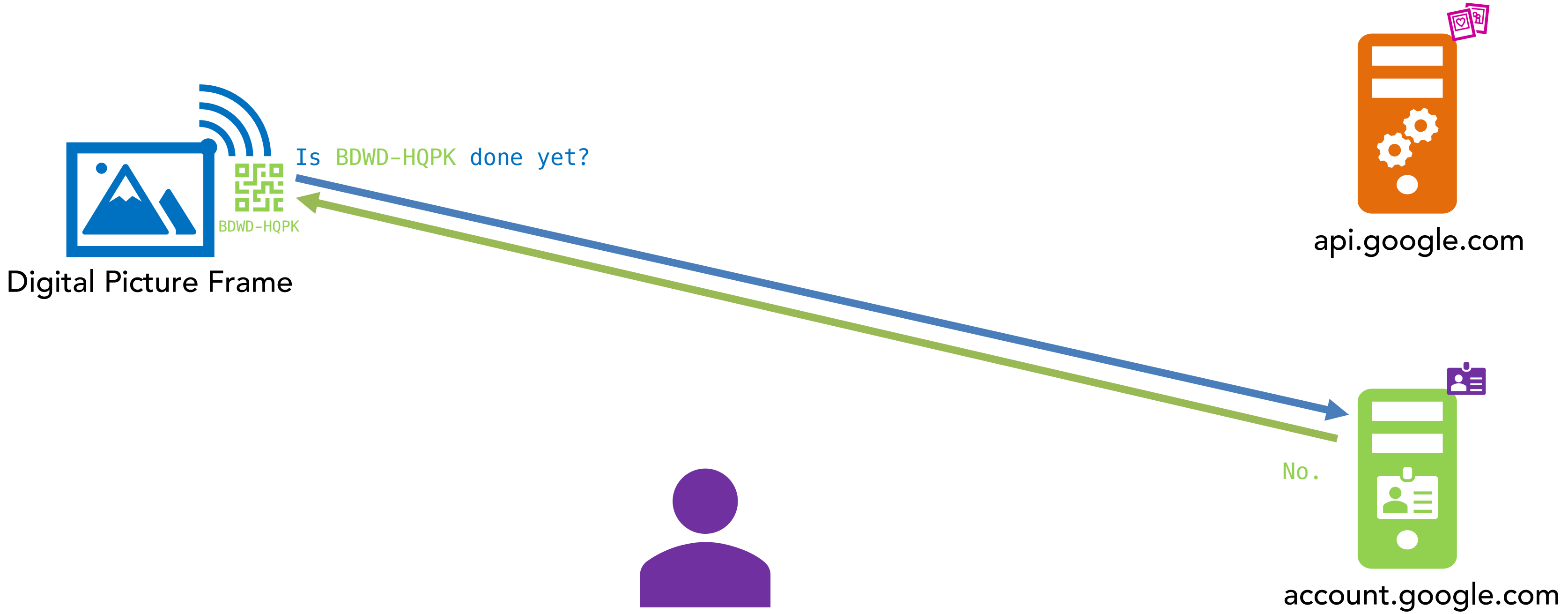
OAuth 2.0 Flows – Device Code



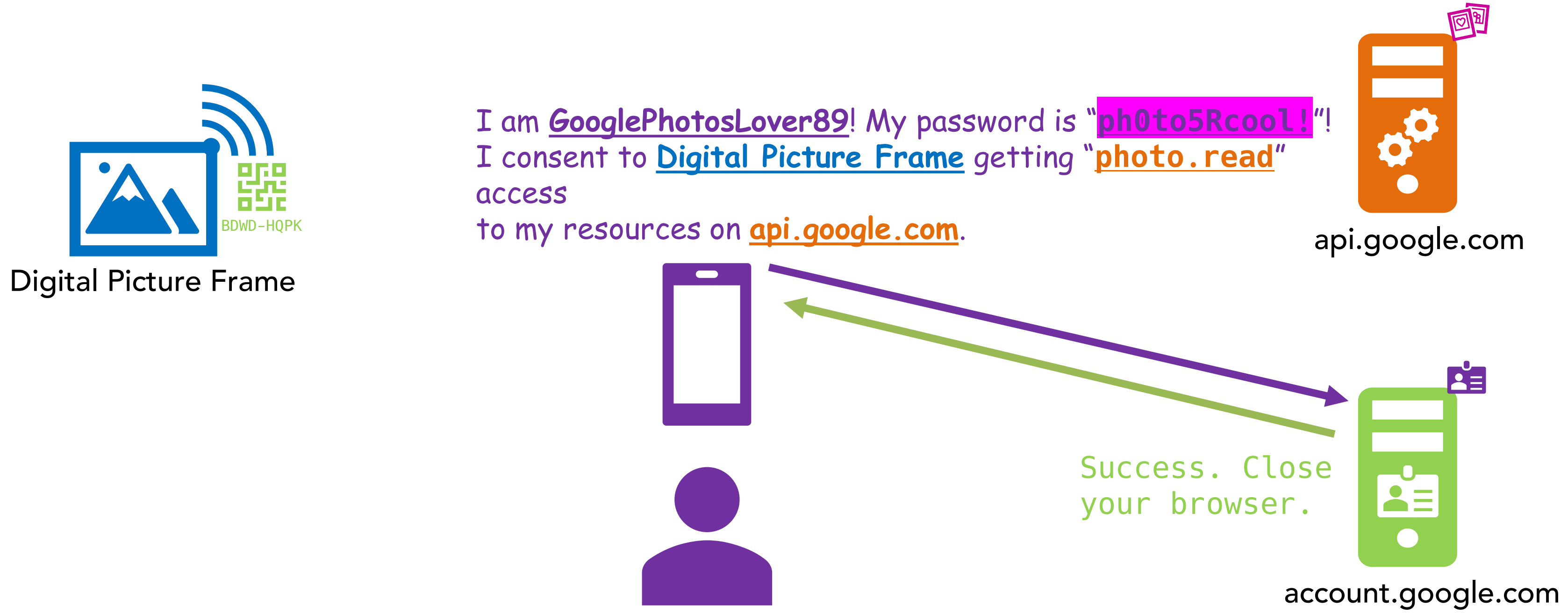
OAuth 2.0 Flows – Device Code



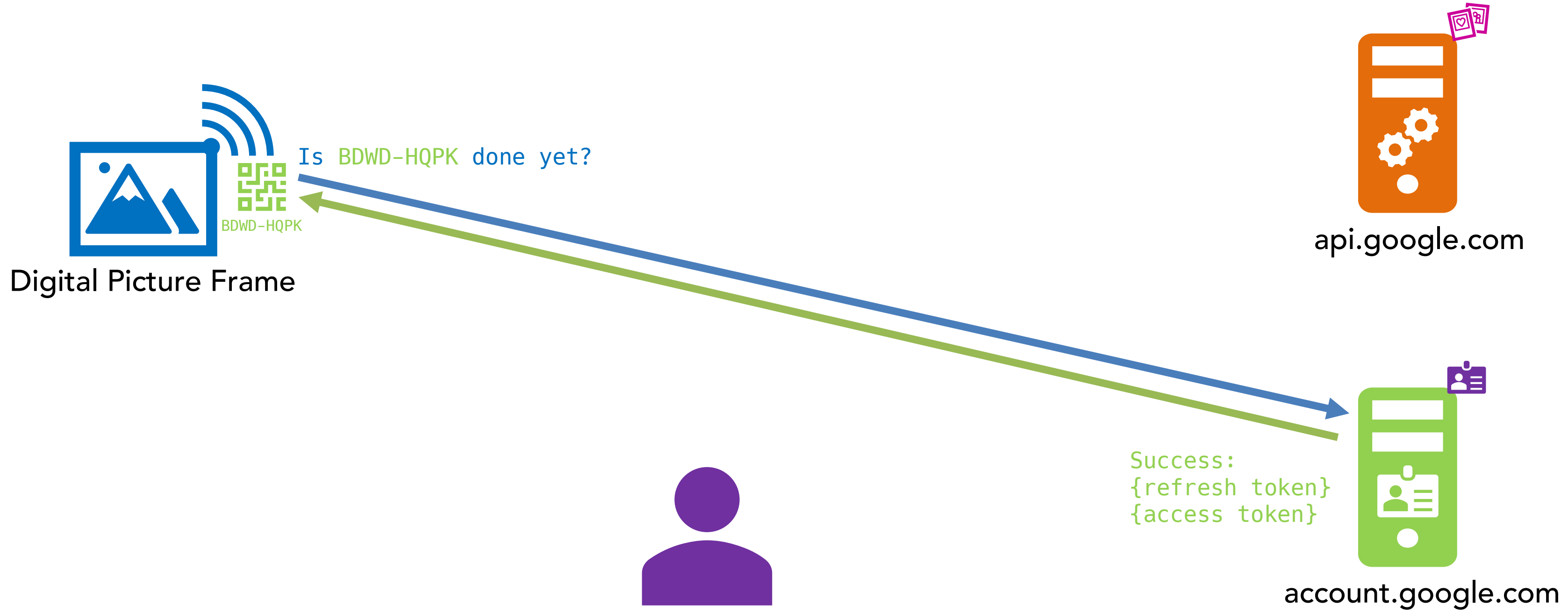
OAuth 2.0 Flows – Device Code



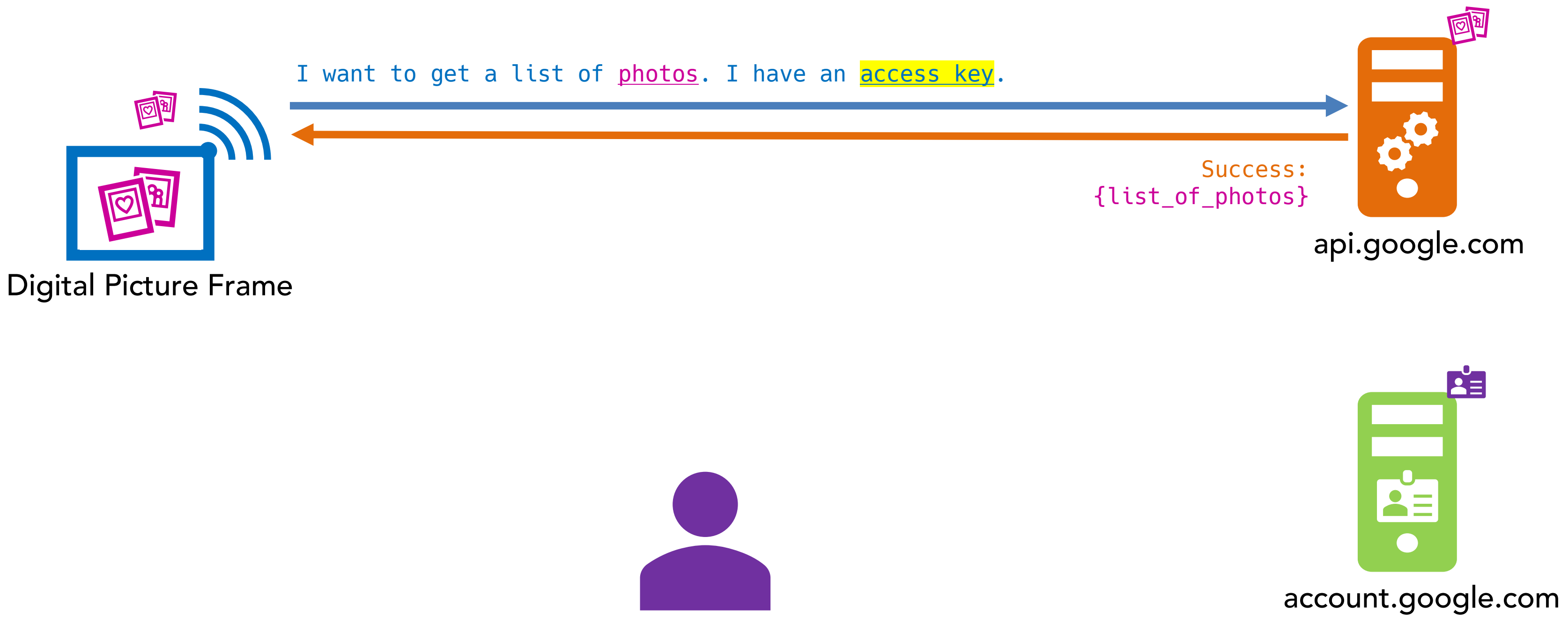
OAuth 2.0 Flows – Device Code


BDWD-HQPK
Digital Picture Frame
I am GooglePhotosLover89! My password is "ph0to5Rcool!"!
I consent to Digital Picture Frame getting "photo.read" access
to my resources on api.google.com.
api.google.com
Success. Close your browser.
account.google.com

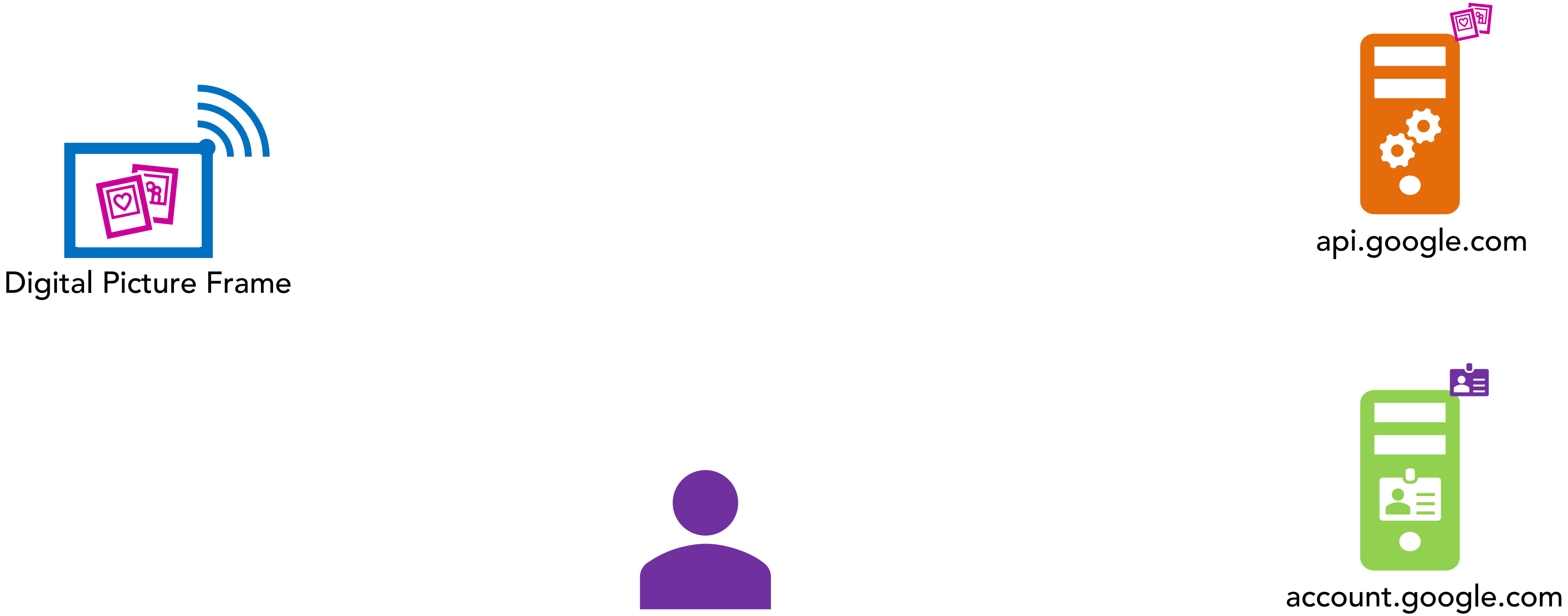
OAuth 2.0 Flows – Device Code

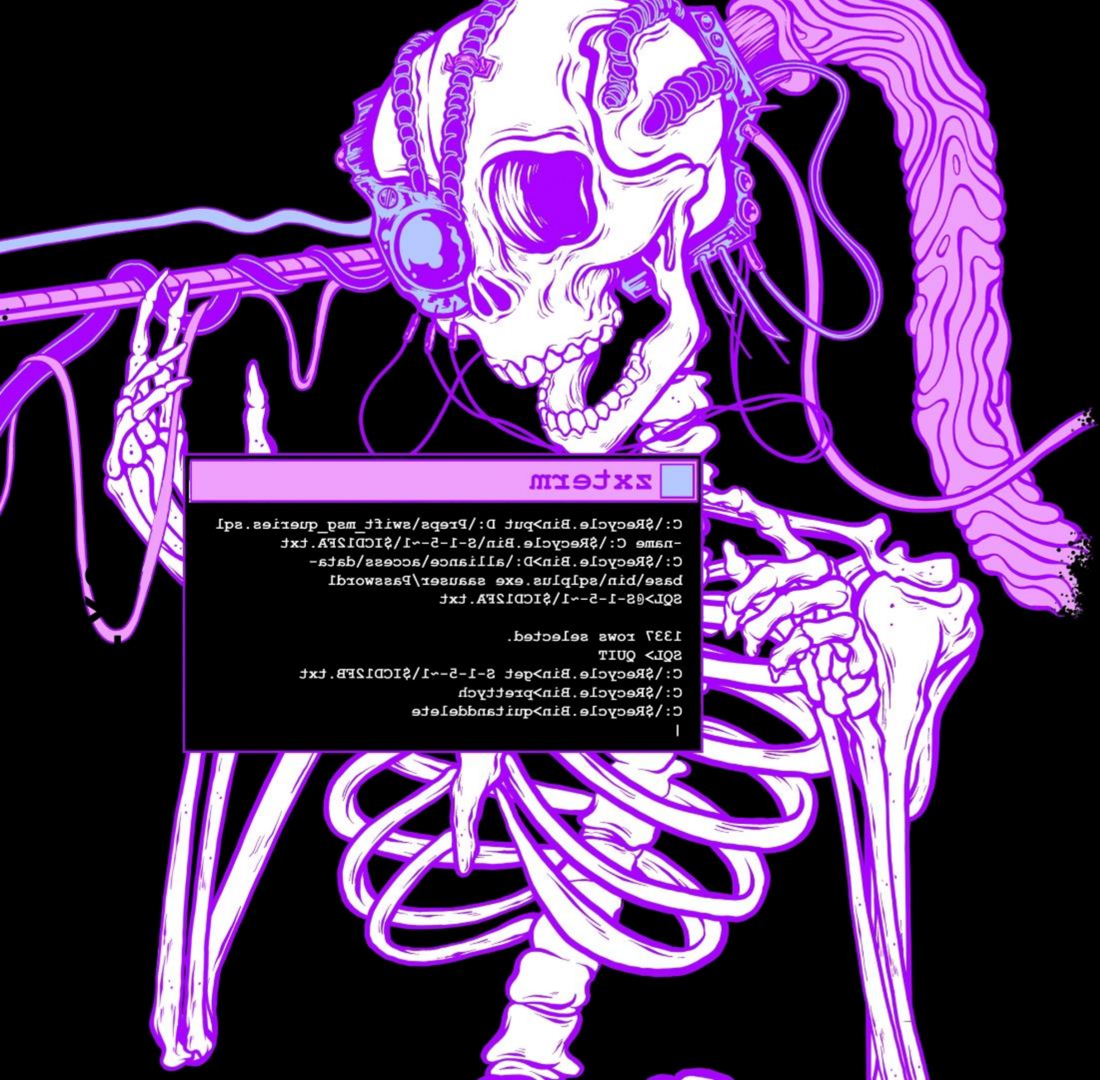


OAuth 2.0 Flows – Device Code



OAuth 2.0 Flows – Device Code





```
C:/$Recycle.Bin>put D:/Preps/swift_msg_queries.sql
-name C:/$Recycle.Bin/S-1-5-~1/$ICD12FA.txt
C:/$Recycle.Bin>D:/alliance/access/data-
base/bin/sqlplus.exe saauser/Password1
SQL>@S-1-5-~1/$ICD12FA.txt

1337 rows selected.
SQL > QUIT
C:/$Recycle.Bin>get S-1-5-~1/$ICD12FB.txt
C:/$Recycle.Bin>prettych
C:/$Recycle.Bin>quitanddelete
|
```

Questions?

Twitter: @mattcotterellnz
Email:
matt.cotterell@zxsecurity.co.nz