



Haojian Jin

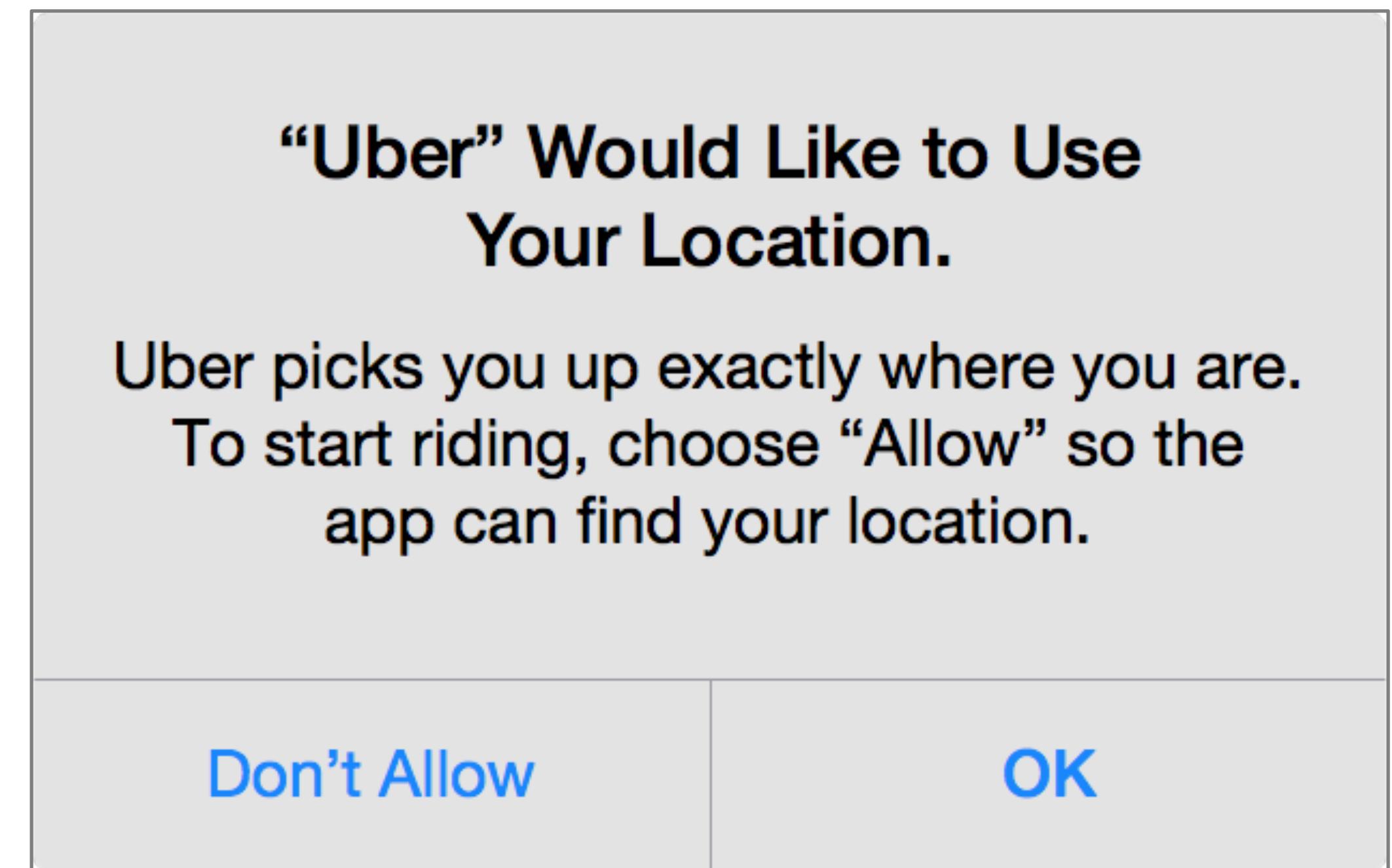
Logistics

1. New review format.
2. Discussions
 1. Presentations & Peer evaluation.
 2. Ask me anything.
3. Final project
 1. Abstract due Feb. 15.
 2. Final report due Mar. 19
4. Grades

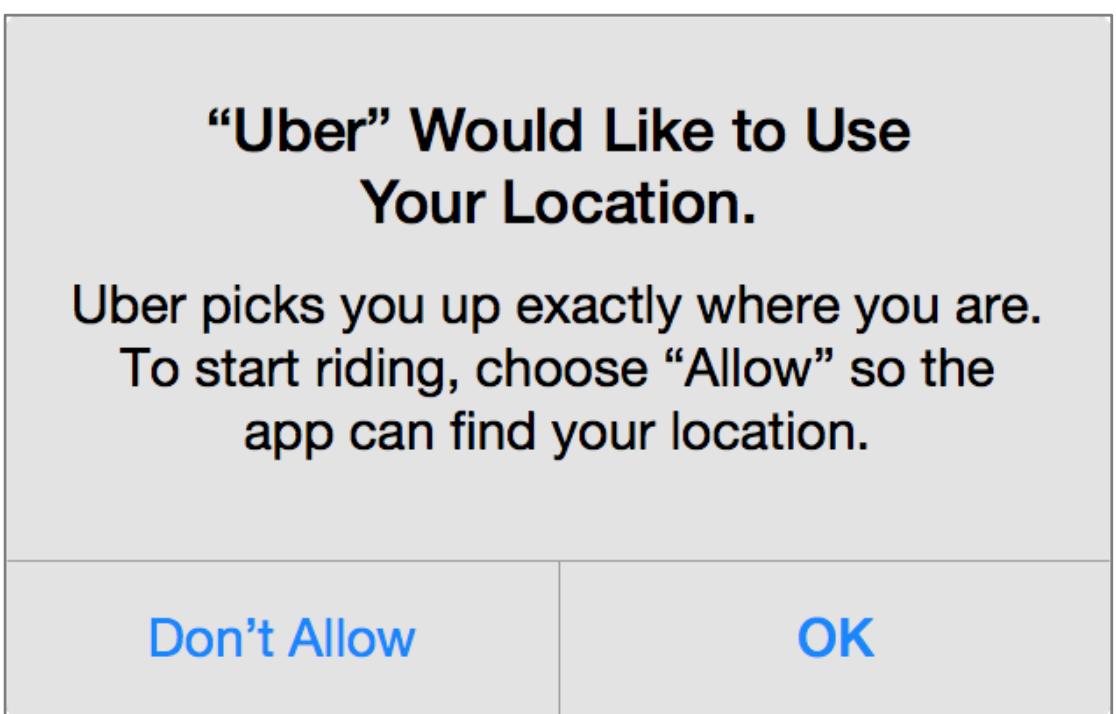
Recap: Location Privacy

- Why location privacy?
- Location-based applications
- Locating technologies
- Protecting location privacy
- Beyond location privacy
- Purpose framework
 - Data collection, data processing, data usages

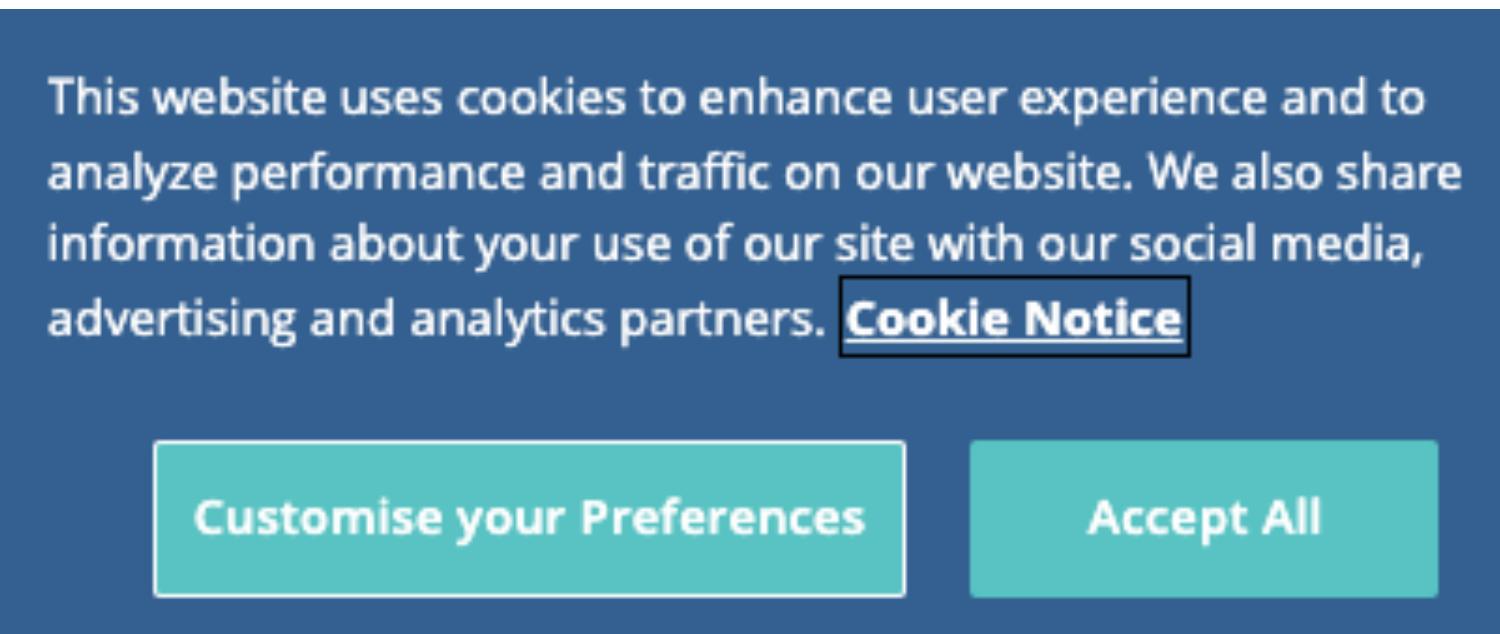
Todays' topic: Permissions



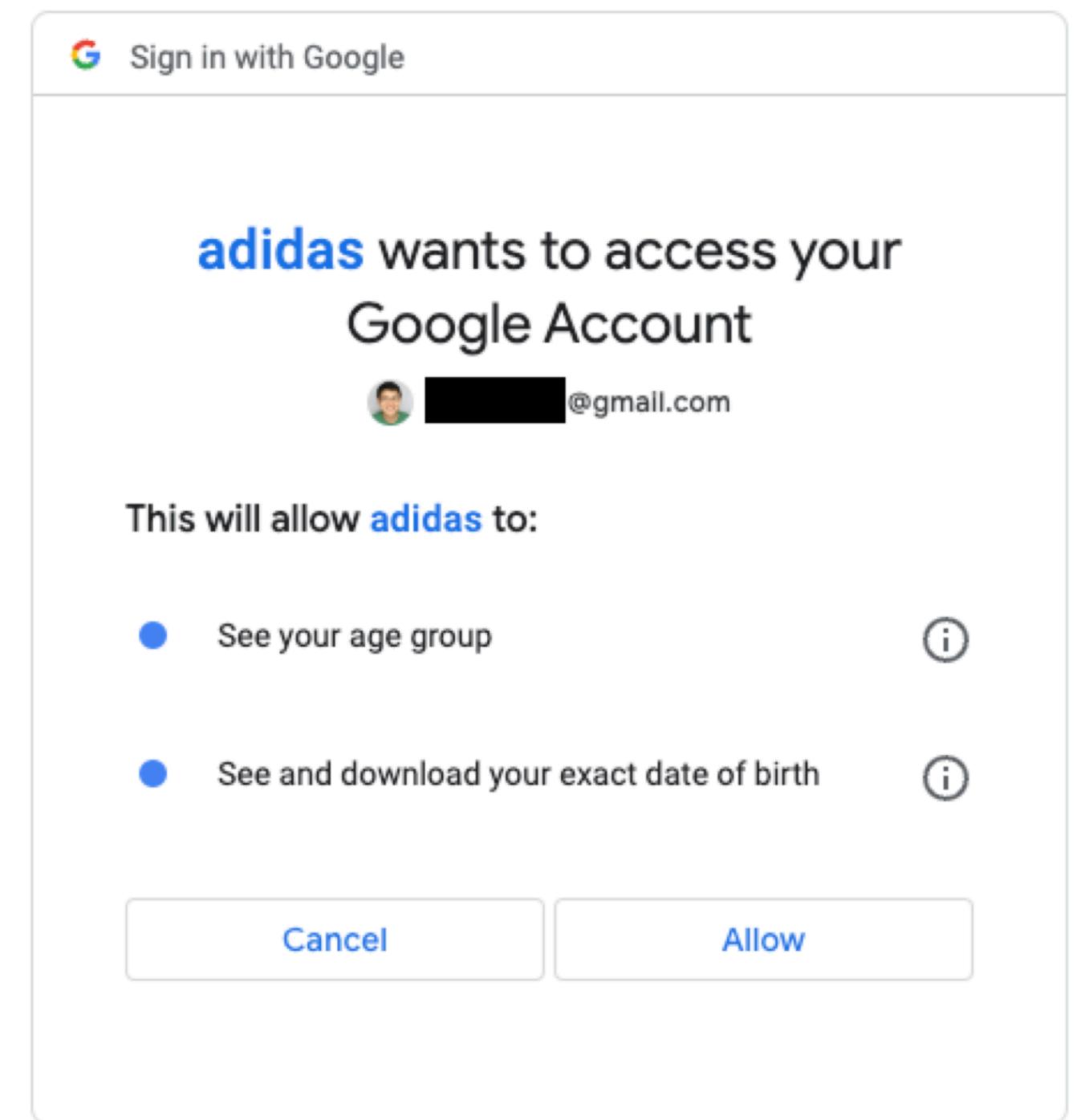
Three systems



Android permission

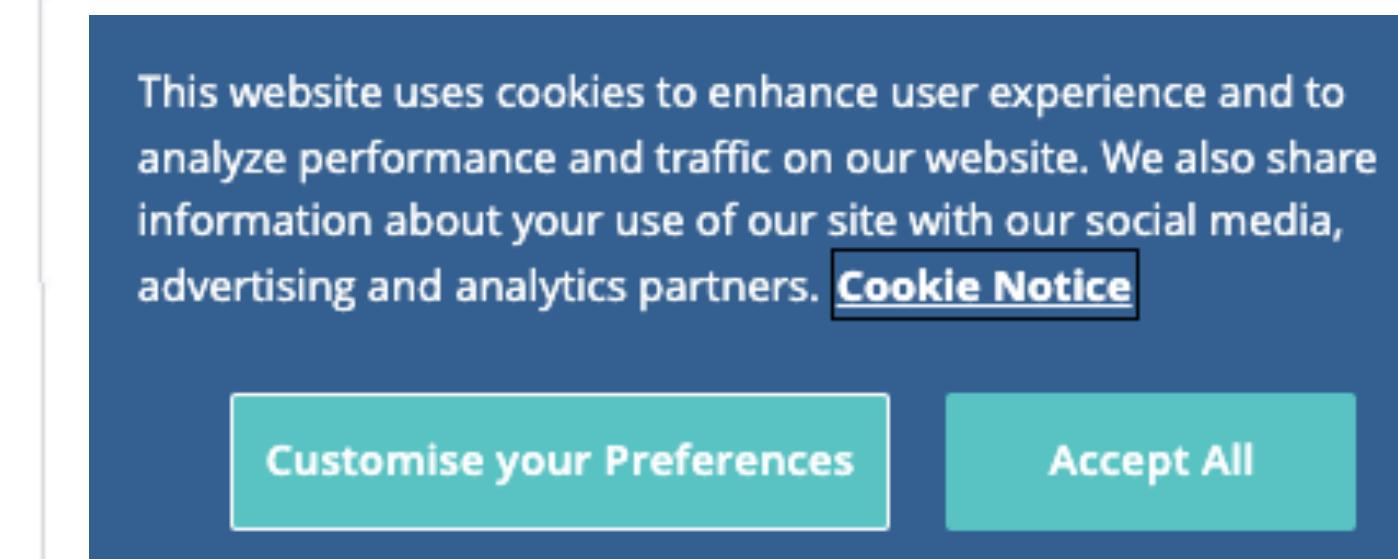
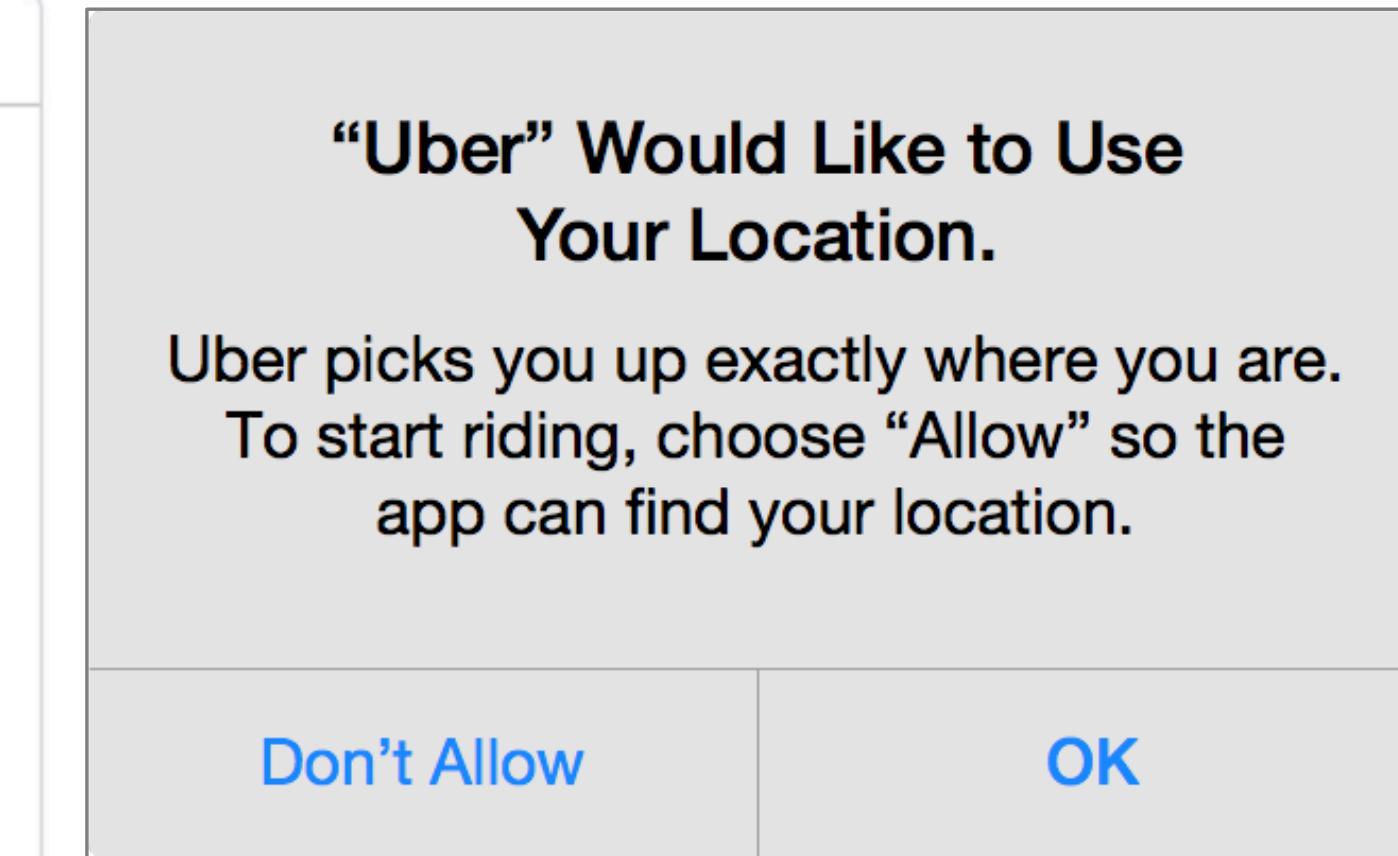
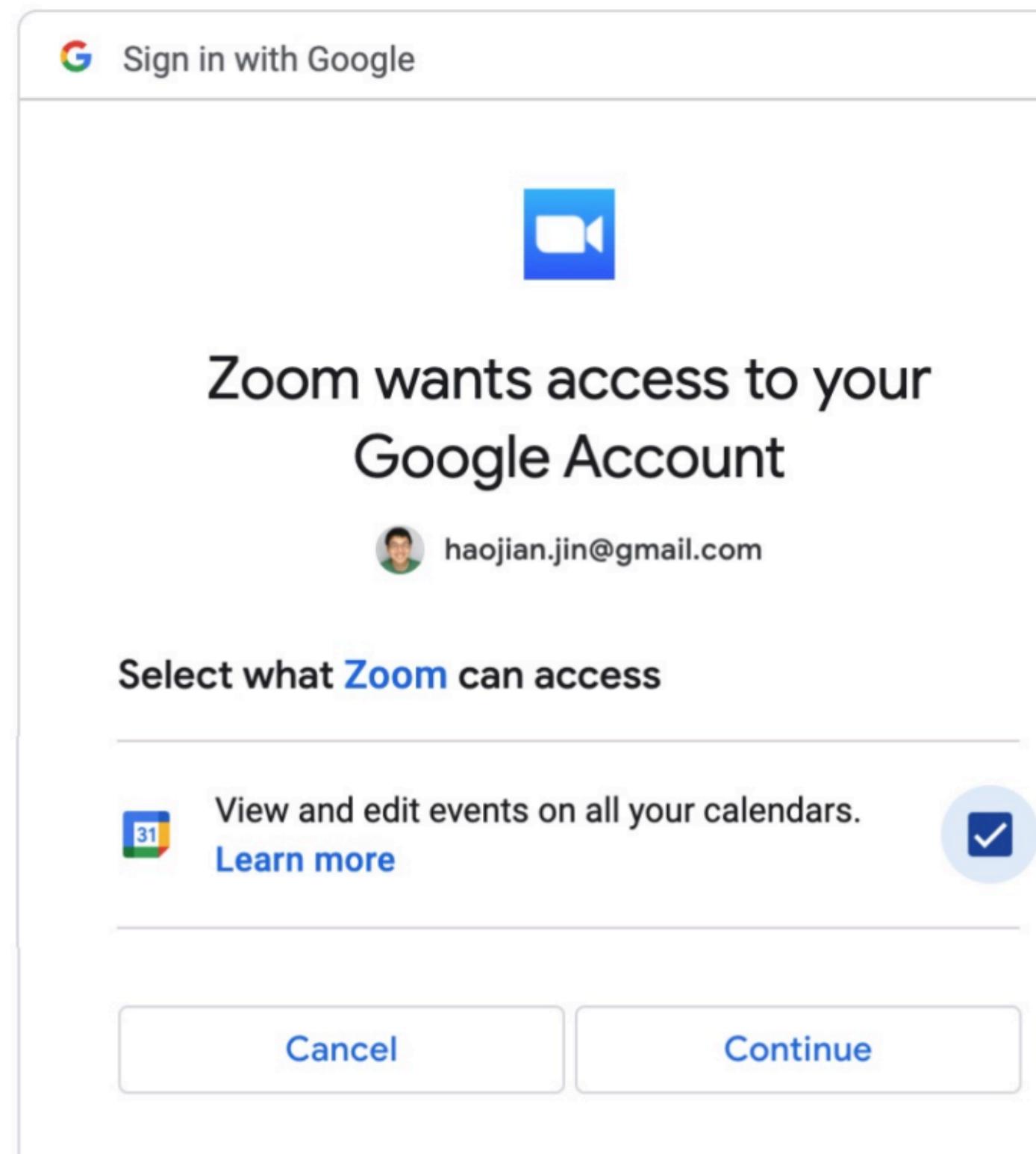


Browser cookie consent



OAuth

Three systems

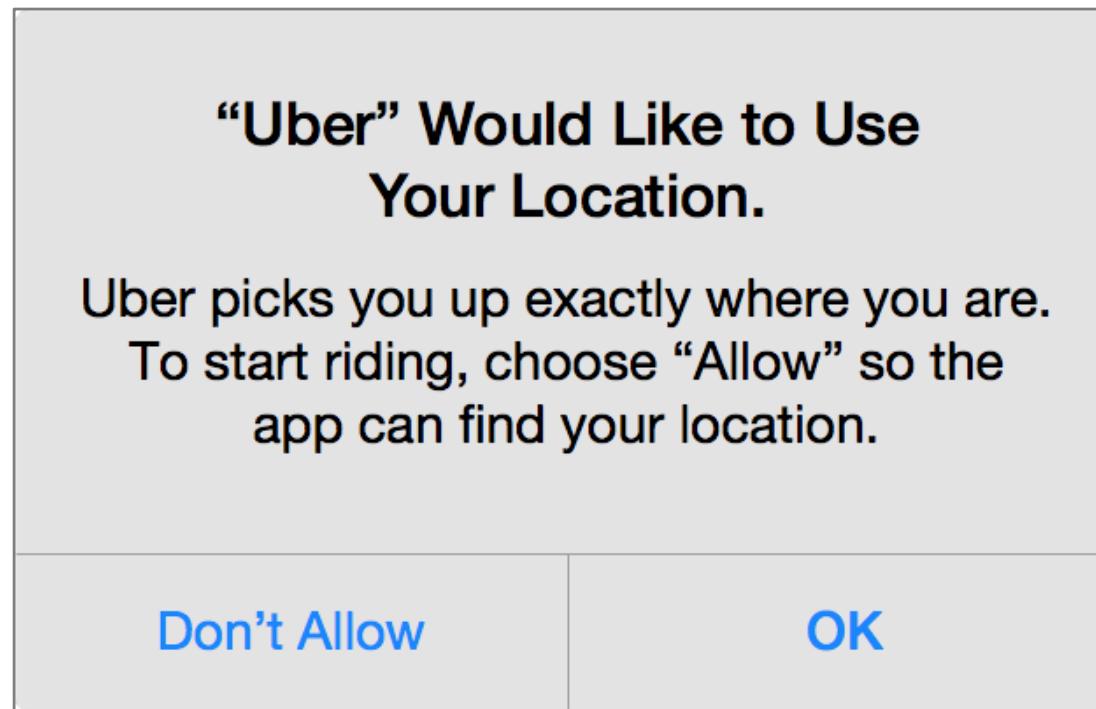


Android permission

Browser cookie consent

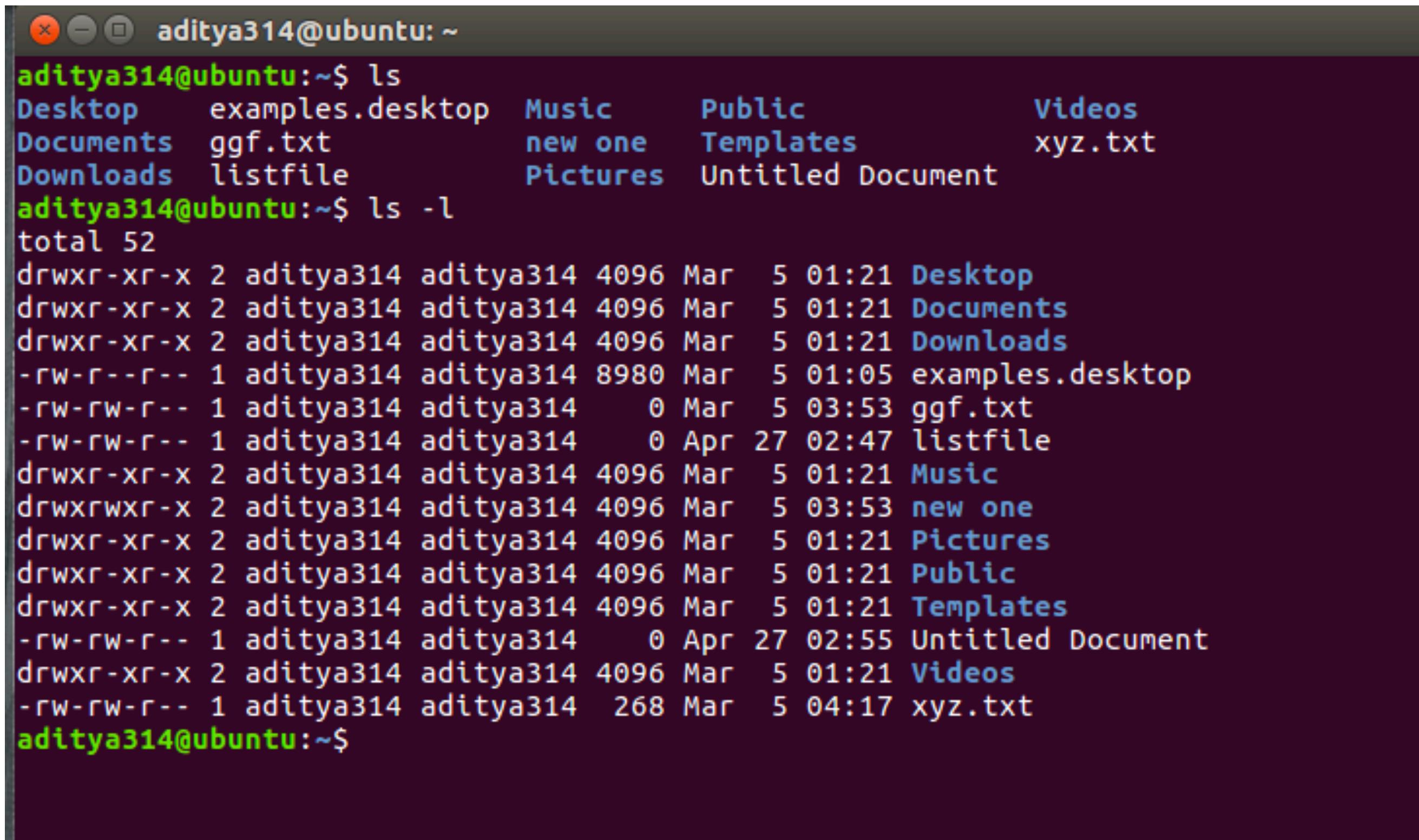
OAuth

Key privacy concepts



- Control
- Notice
- Consent
- Usability
- System mechanisms

Linux systems



A screenshot of a terminal window titled "aditya314@ubuntu:~". The window displays two commands: "ls" and "ls -l". The "ls" command shows a directory structure with files like Desktop, Documents, Downloads, examples.desktop, ggf.txt, listfile, Music, new one, Pictures, Public, Templates, Videos, Untitled Document, and xyz.txt. The "ls -l" command provides detailed file permissions and metadata for each item.

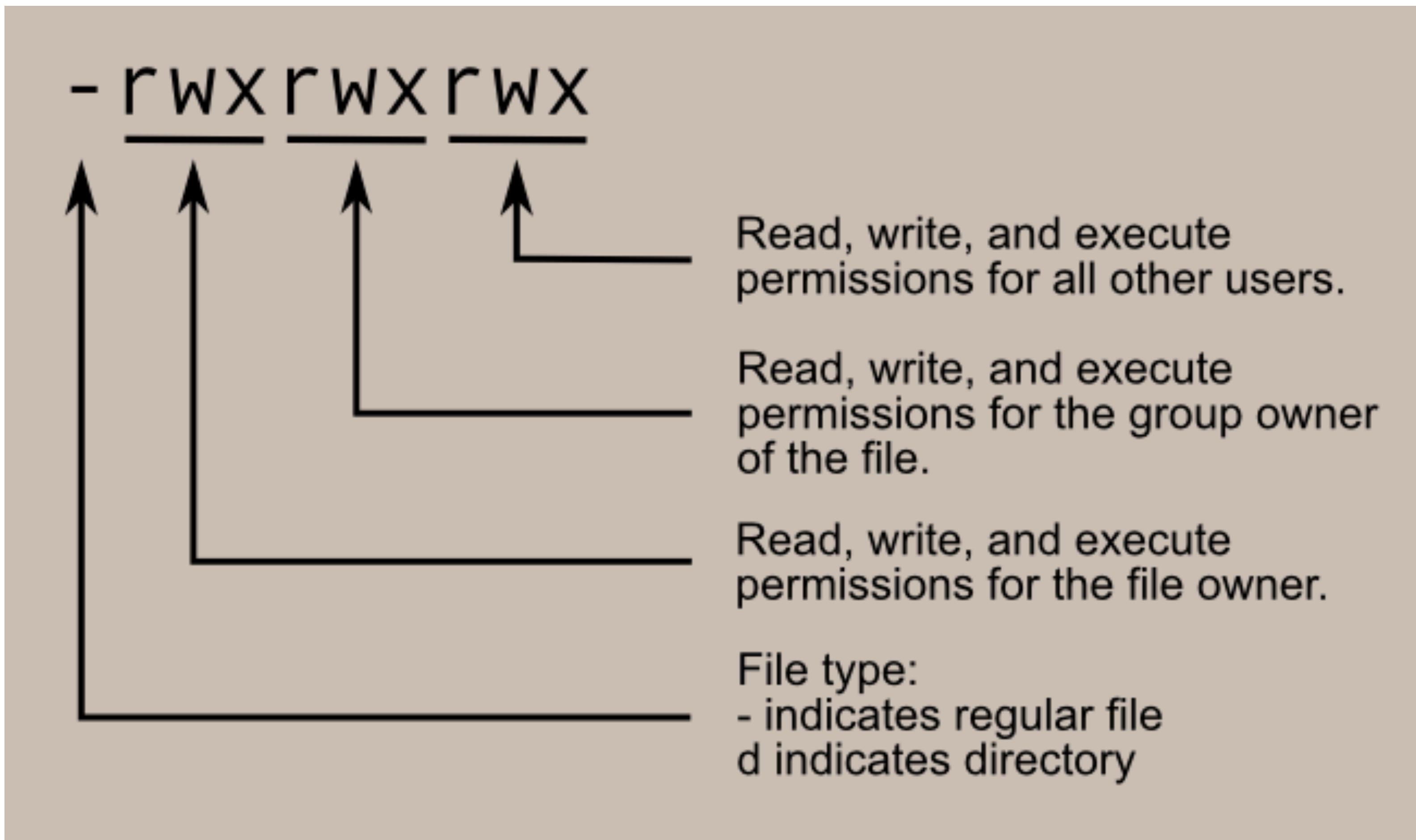
```
aditya314@ubuntu:~$ ls
Desktop  examples.desktop  Music      Public      Videos
Documents  ggf.txt        new one    Templates   xyz.txt
Downloads  listfile       Pictures   Untitled Document

aditya314@ubuntu:~$ ls -l
total 52
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Desktop
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Documents
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Downloads
-rw-r--r-- 1 aditya314 aditya314 8980 Mar  5 01:05 examples.desktop
-rw-rw-r-- 1 aditya314 aditya314     0 Mar  5 03:53 ggf.txt
-rw-rw-r-- 1 aditya314 aditya314     0 Apr 27 02:47 listfile
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Music
drwxrwxr-x 2 aditya314 aditya314 4096 Mar  5 03:53 new one
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Pictures
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Public
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Templates
-rw-rw-r-- 1 aditya314 aditya314     0 Apr 27 02:55 Untitled Document
drwxr-xr-x 2 aditya314 aditya314 4096 Mar  5 01:21 Videos
-rw-rw-r-- 1 aditya314 aditya314  268 Mar  5 04:17 xyz.txt

aditya314@ubuntu:~$
```

- Users
- Permissions

File permissions



Why permissions?



- Notice
- Control
- Consent

Why permissions?



- Notice
 - Did users read the form?
 - Did users understand the form?
 - Can users understand the form?
 - Is the form accurate?

Why permissions?



- Control
 - Do they have a control?
 - Is it all-or-nothing?
 - Is it easy to control?
 - Do developers respect users' control?

Why permissions?

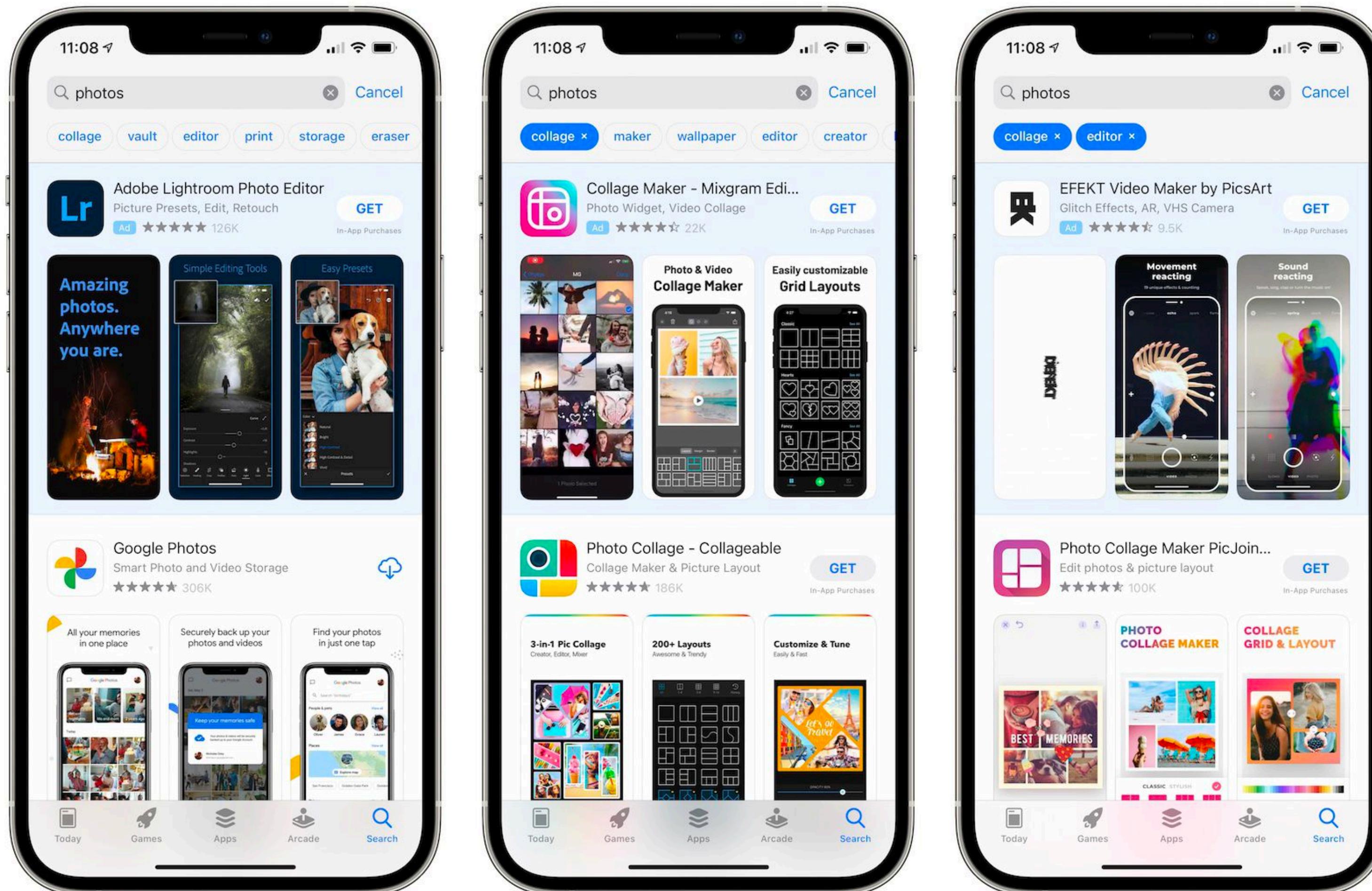


- Consent
 - What's valid consent?
 - What if I agree to give up my privacy?
 - What if I leak other people's privacy?

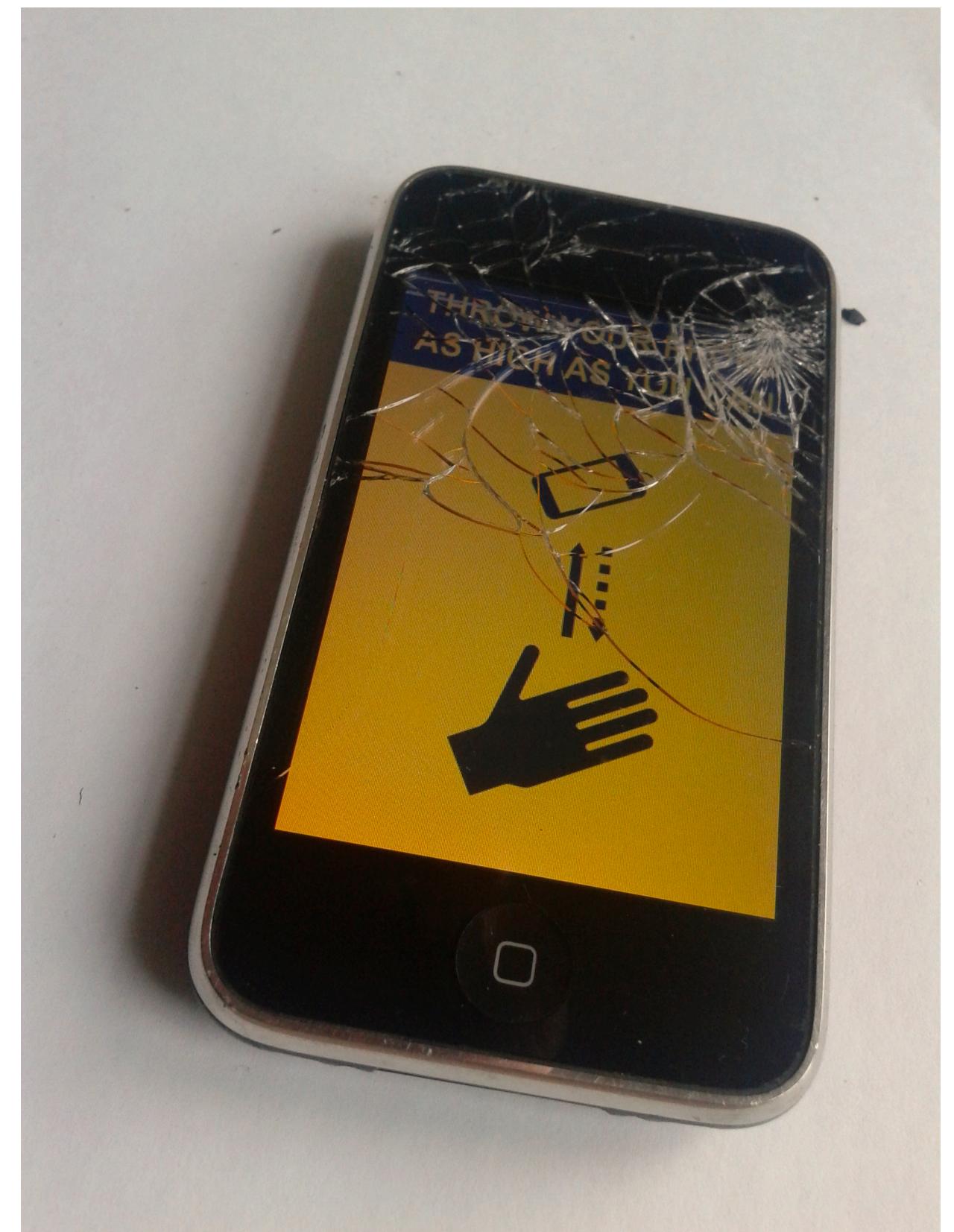
What works in med do not work in privacy.

- Scale
 - Burdens on users
 - Intangible data
 - Hard to enforce
 - Motivations, awareness, knowledge
 - Less sensitive on long-term risks

App Store



S.M.T.H.: Send Me to Heaven



100+ million => a few billion

- Putting apps in a sandbox & permissions.
 - Developers cannot ask/persuade/trick users for permissions to do dangerous thing.
 - Better way to distribute software
 - Centralized payments

How Android Applications work (1)

- Java source code → compiled into .dex byte-code file
- .dex file + Manifest file + resources = .apk archive
- Application isolation → **system level security**
 - Linux process, address space
 - VM (Dalvik Virtual Machine) for each application
 - unique Linux user ID
 - direct access only to its own data
 - API-based access to other apps' resources
- Not a single entry-point (no main)

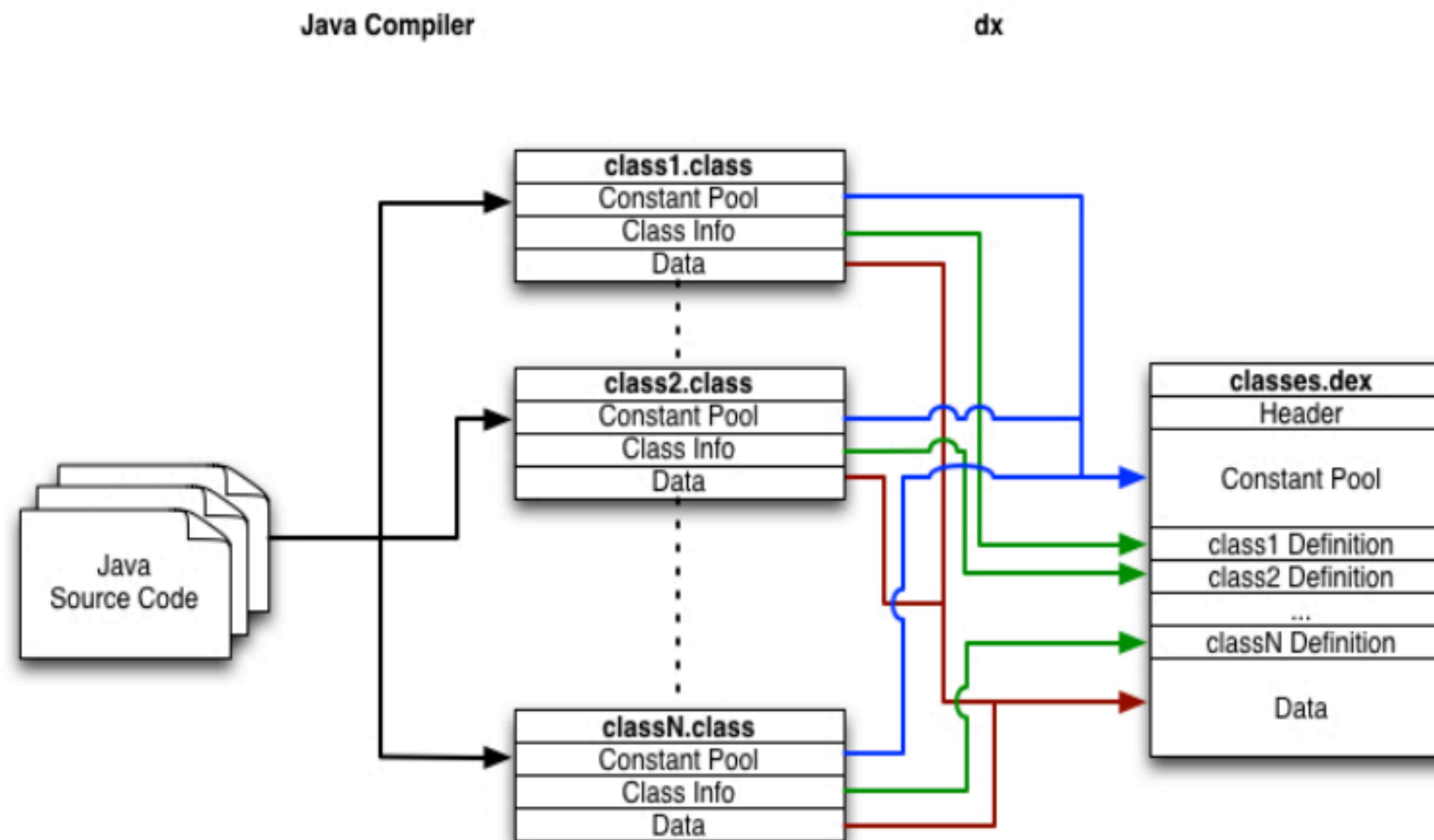
How Android Applications work (2)

- Applications can start each other
- Based on Components and Intents
- .dex - Dalvik Executable format
- Dalvik is optimised for mobile architectures
 - low memory consumption
 - Dex results in smaller binaries than JAR

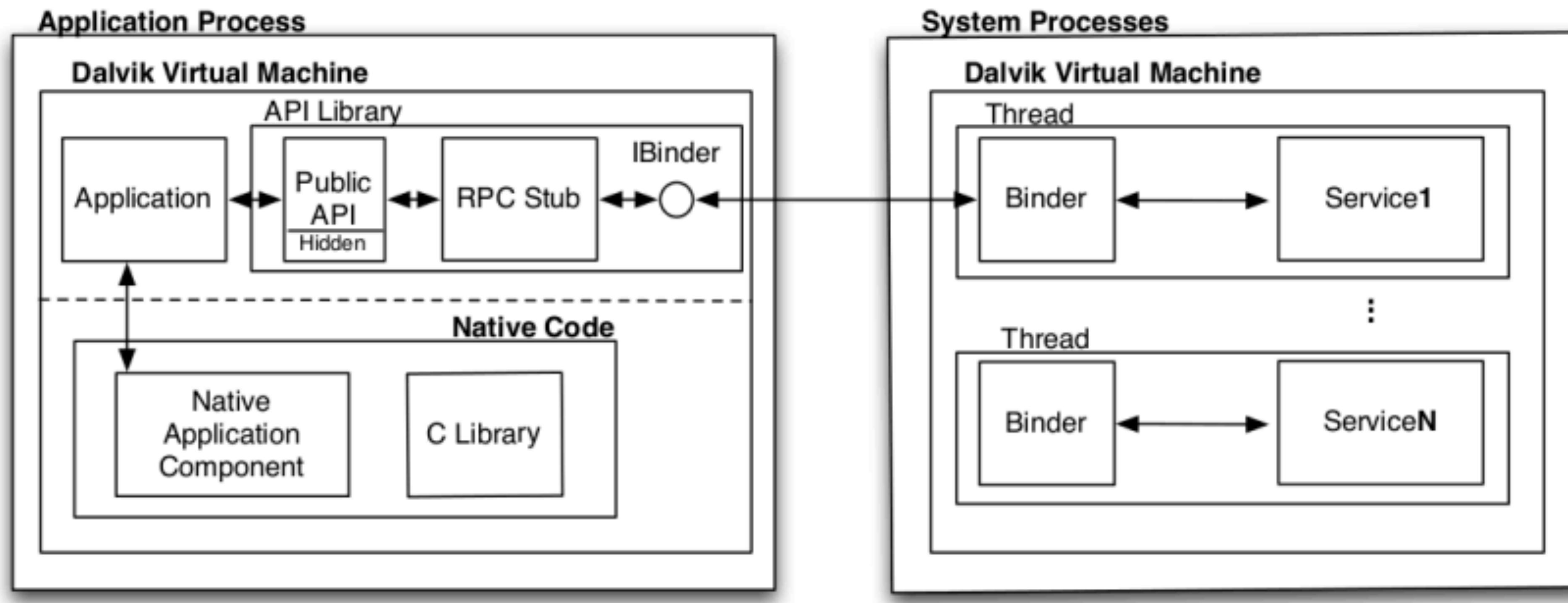
How Android Applications work (3)

- register-based architecture (JVM is stack-based)
 - Java VM cannot execute Dalvik code
 - 16-bit instructions
 - copy-on-write memory sharing
 - dx cross-compiler - works with javac output

Compiling applications



Android architecture



AndroidManifest

- XML configuration file, Every application must have it
- Contains:
 - application's name, icon, labels
 - linked libraries
 - application components: <activity>, <service>, <receiver>, <provider> tags
 - Activity shown at launch time
 - Intent filters
 - **Permissions**

AndroidManifest Example

Panoramio App:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.google.android.panoramio">
    <application android:icon="@drawable/icon">
        <activity android:name=".Panoramio" android:label="@string/app_name"
            android:theme="@style/Theme.Panoramio">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".ImageList" android:label="@string/app_name"
            android:theme="@android:style/Theme.Light"/>

        <activity android:name=".ViewImage" android:label="@string/app_name"
            android:theme="@style/Theme.Panoramio"/>

        <activity android:name=".ViewMap" android:label="@string/app_name"/>

            <uses-library android:name="com.google.android.maps" />
    </application>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
</manifest>
```

Application Framework Security

- Android Framework Security → coarse-grained control
- Mandatory Access Control(MAC) enforced by middleware

Application Framework Security

- Components protected using access permission labels
 - declared in the `AndroidManifest` file
 - can not be changed after installation
 - 4 protection levels
 - normal - always granted
 - dangerous - requires user approval
 - signature - matching certificate
 - signature or system - matching certificate with system image

Permissions

- At install-time each application requests a list of permission
- All permissions must be granted at install time - all or nothing
- Protect access to Android components, services and APIs
 - e.g API for access to phone's hardware
 - ~130 API-defined permissions in `Manifest.Permissions` class

Permissions

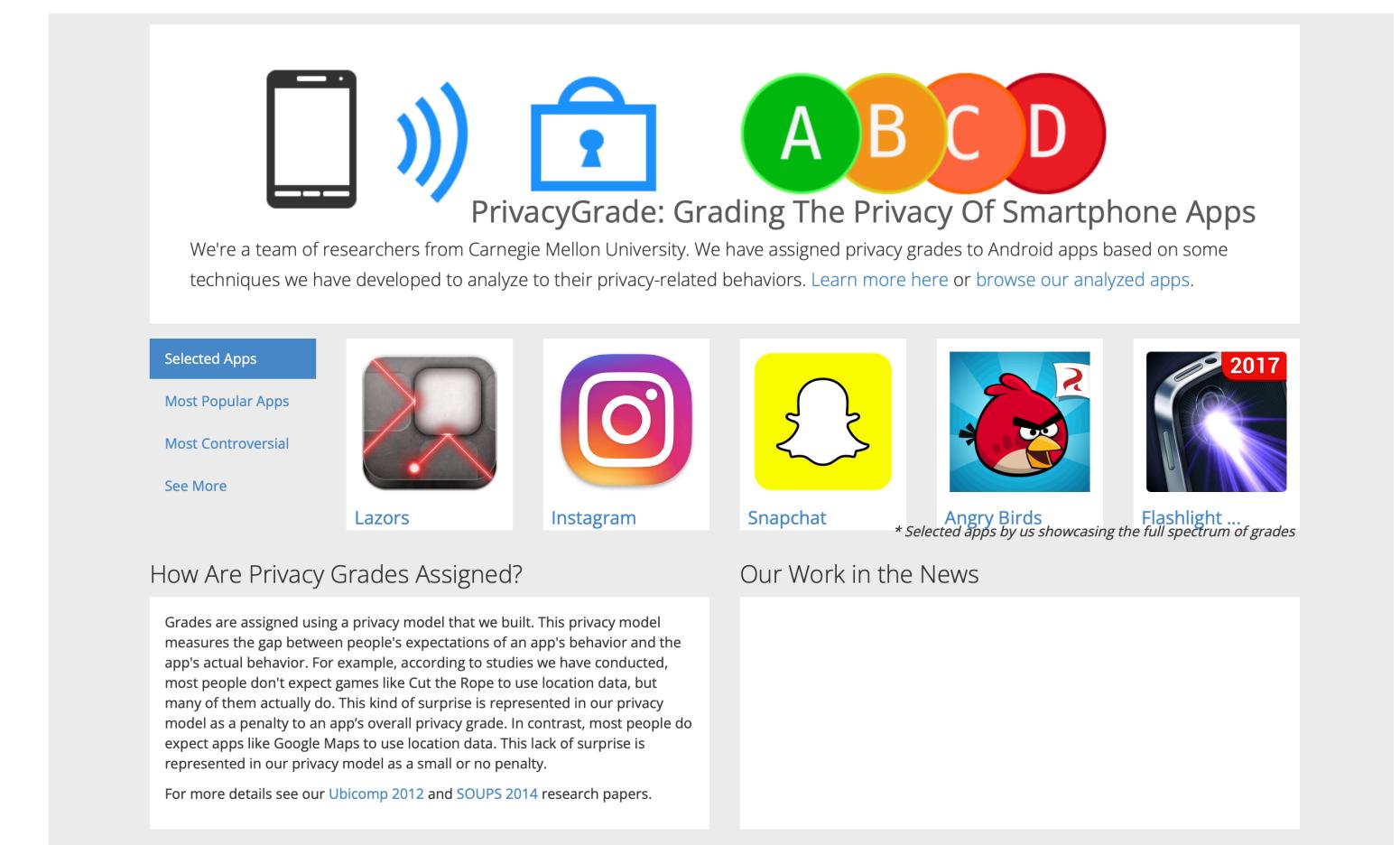
- No custom-defined permissions by developers
 - name conflicts may appear
 - current research on Android permissions doesn't take them into consideration
- PackageManagerService in the middleware checks the permissions for a request

User Attention, Comprehension, and Behavior

- Are users paying attention to the permissions?
- Do users understand the permissions?
- Can users make correct security decisions?
- Results: too few users comprehend or pay attention

Research around Android permissions

- The problem: unnecessary use of permissions
- The proposed solution: static analysis of API calls
- Permission map -
 - identifies permissions for Intents, Content Provides, API calls
 - determines if an app is overprivileged or not



Research methods for Android Permissions

- Static analysis:
 - Map of permissions for each method in the Android API
- Dynamic analysis:
 - Log permission checks - modified middleware
- Blackbox test:
 - test cases for API calls, Intents, Content Providers

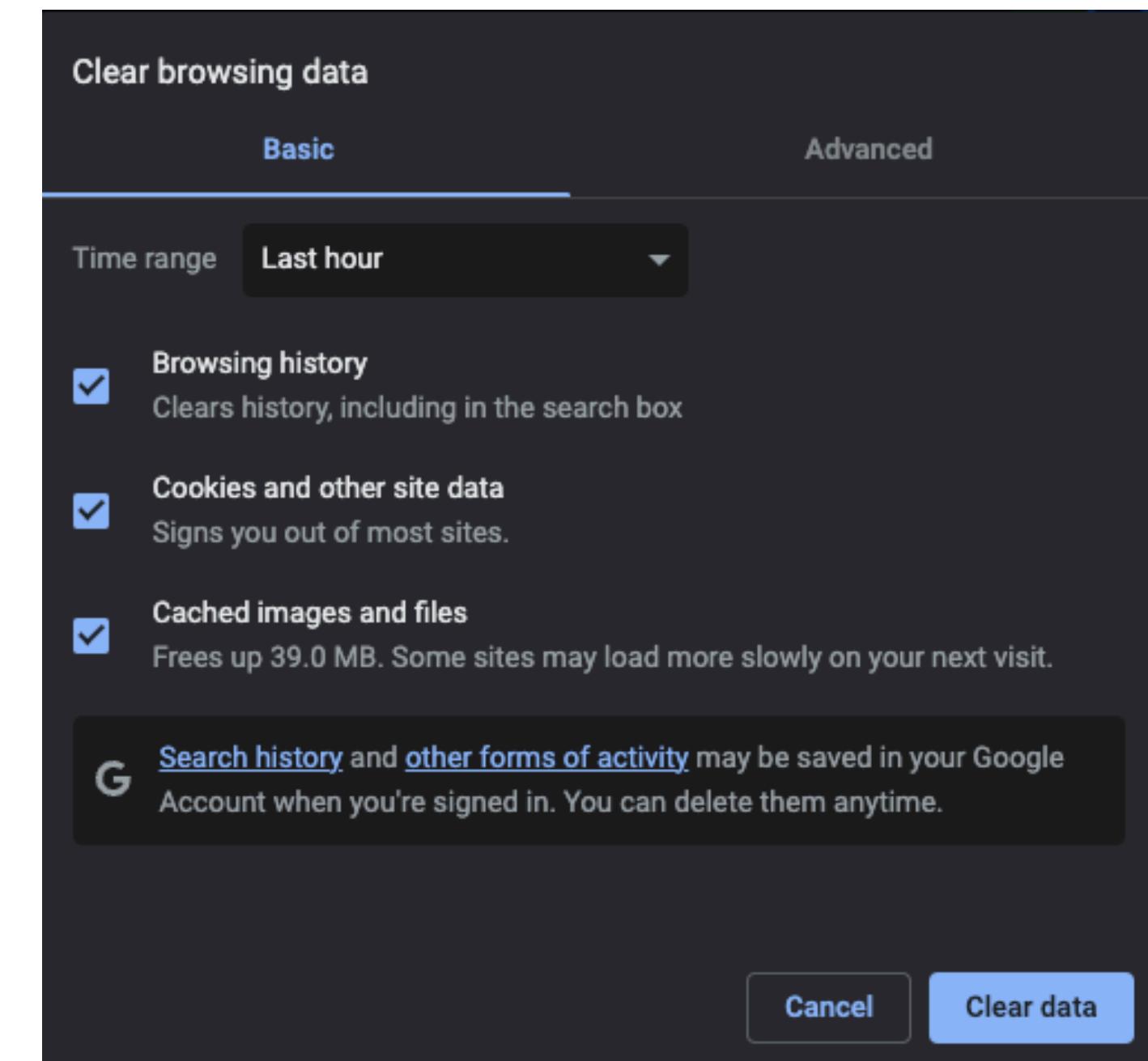
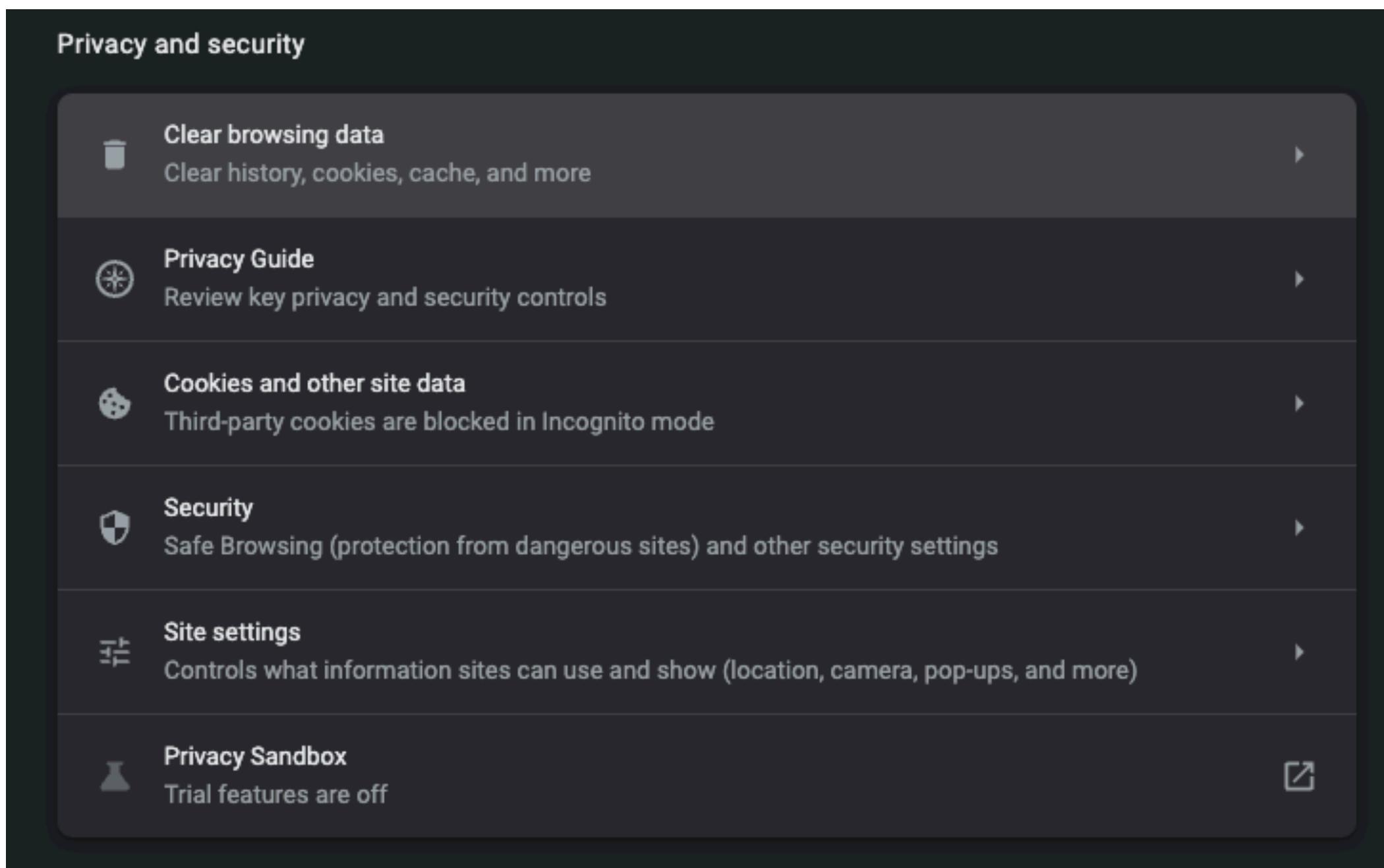
Cookie consent banner

This website uses cookies to enhance user experience and to analyze performance and traffic on our website. We also share information about your use of our site with our social media, advertising and analytics partners. [**Cookie Notice**](#)

[Customise your Preferences](#)

[Accept All](#)

What is Cookie?

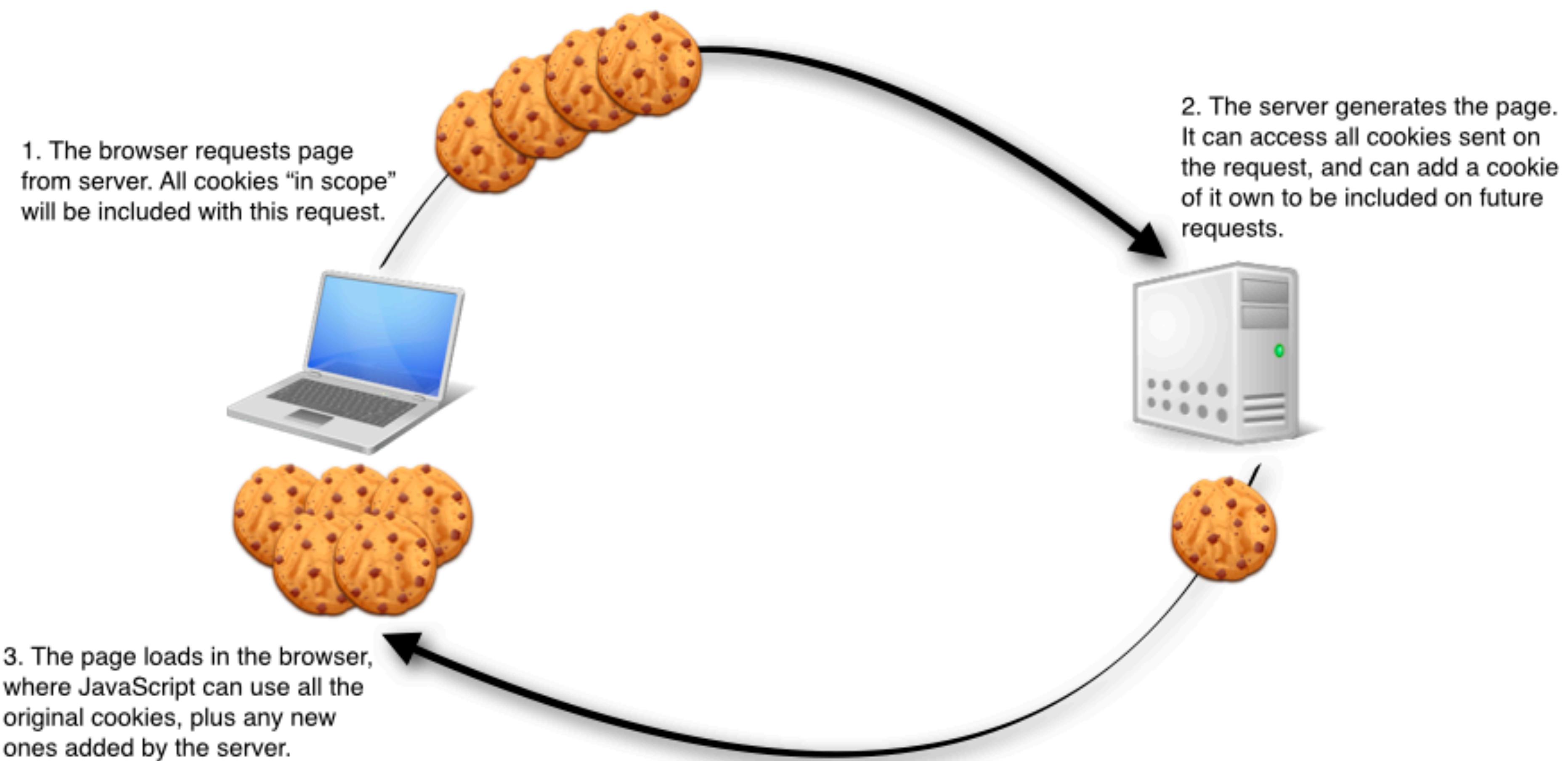


Cookie applications

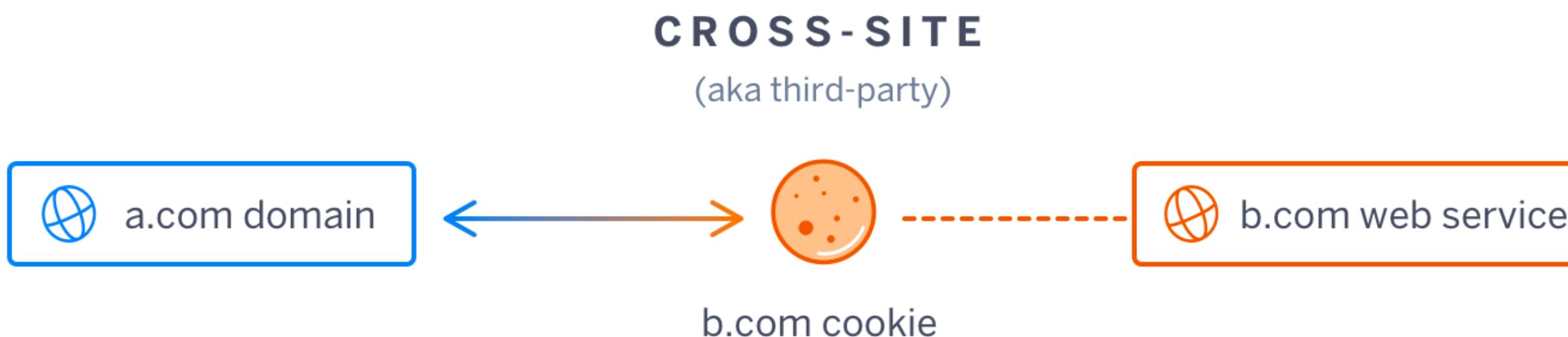
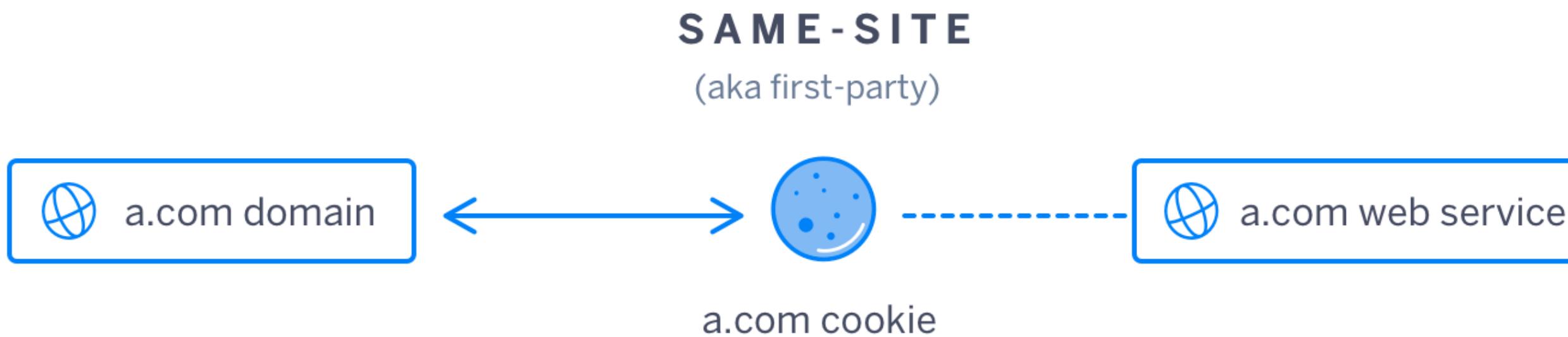


- Login status
- Signout personalization
- Anti-fraud
- Advertisement
- Marketing measurement
- ...

Cookie: stateless to stateful

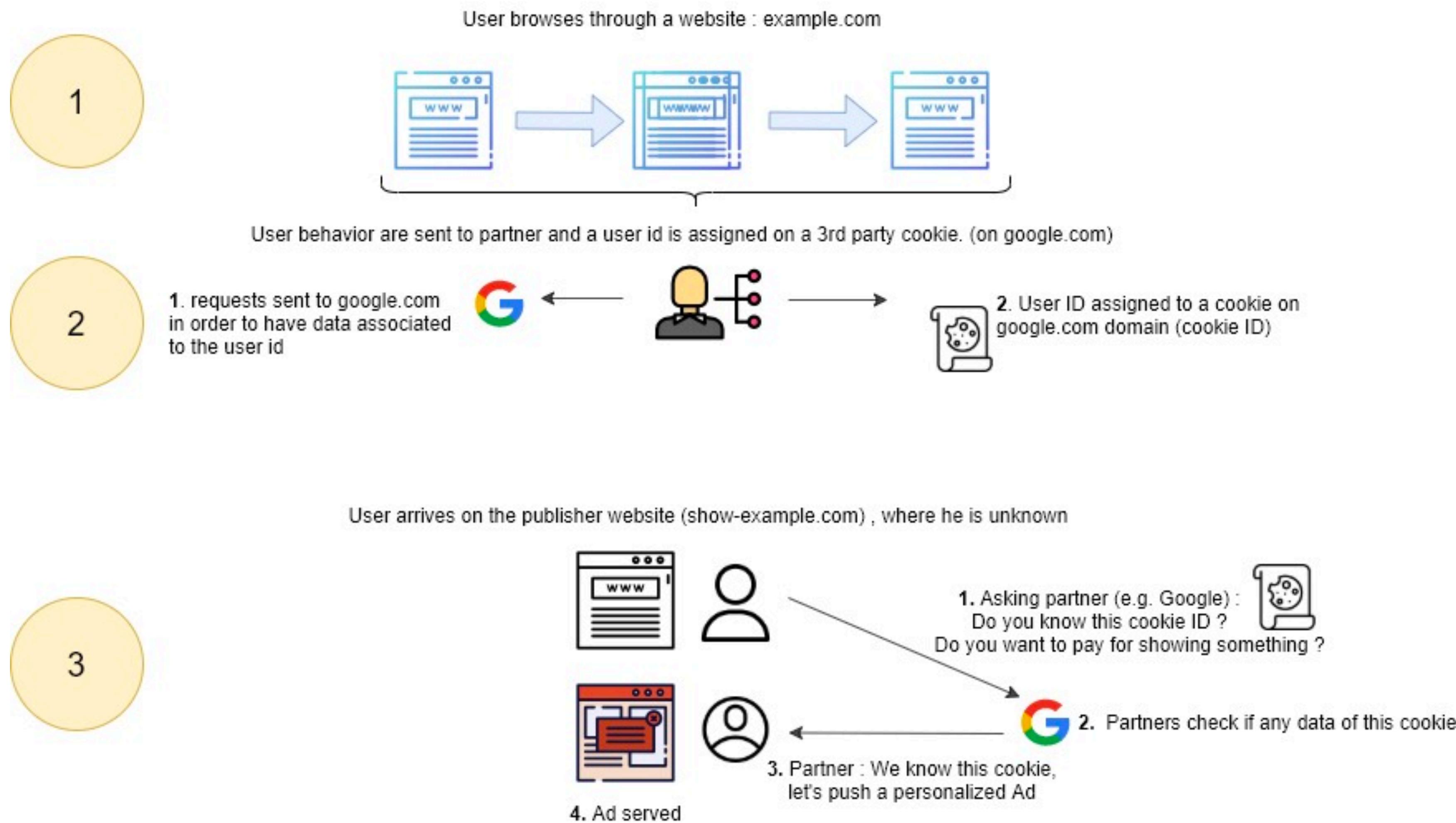


SameSite Cookie (Feb. 2020)



Starting February 4, 2020, Google Chrome will stop sending third-party cookies in cross-site requests unless the cookies are secured and flagged using an IETF standard called **SameSite**.

How Advertisements work?



Required by Multiple Laws

Cookie Banner Guidelines for Each Global Privacy Regulation

In this article, we'll look at cookie banner best practices according to [GDPR](#), [CCPA](#), [ICO](#), [CNIL](#), [LGPD](#), and the Nevada Privacy Law ([SB-220](#)) to help websites stay compliant.

Last Updated: August 5, 2022

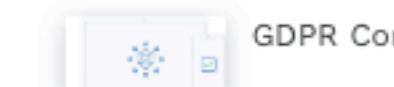
- [GDPR Cookie Banner](#)
- [CCPA Cookie Banner](#)
- [ICO Cookie Banner](#)
- [CNIL Cookie Banner](#)
- [LGPD Cookie Banner](#)
- [Nevada Privacy Law Cookie Banner](#)
- [Conclusion](#)

01 GDPR Cookie Banner

The **General Data Protection Regulation (GDPR)** is the privacy regulation in force in the EU. The regulation went into effect on May 25, 2018, so many website owners are already familiar with GDPR cookie banner requirements.

One of the central protections GDPR gives EU citizens is that they have the right to be informed when businesses collect data about them. Businesses must let individuals know why they are collecting the data, how long they keep the data for, and which organizations they will share the data with. Individuals also have the right to object to the processing of their personal data in some

You Might Also Like



GDPR Con

GDPR Banner requirements

- Include a Button to Accept Cookies
- Provide Detailed Information About Cookie use
- Alert the users if the website shares data with third parties
- Link to the website's cookie policy
- Include a link to the cookie settings

GQ

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners. You may read more about any of the purposes or vendors that we use by clicking 'Show Purposes'. This preference center is accessible at any time through the 'Manage Privacy Preferences' button located on every page.[View Cookie Policy](#) [List of Partners \(vendors\)](#)

ACCEPT

SETTINGS

How developers implement them?

A screenshot of a Google search results page. The search query is "open source cookie banner". The results show several entries, starting with an advertisement for Osano, followed by links to Osano's website, GitHub repository, and a detailed description of the cookie consent solution.

Google open source cookie banner

All Images Shopping News Videos More Tools

About 128,000,000 results (0.45 seconds)

Ad · https://www.osano.com/cookie-consent

Osano: Simple Cookie Banner - Used by 750,000 Companies

"Cookie consent and website privacy made easy. Seamless implementation, fast support."

Osano is an easy-to-use tool that manages all your cookie consent dialogs. Try for free.

Cookie Consent Management
The World's Most Popular Consent Management Platform

Vendor Risk Monitoring
Osano Attorneys Have Reviewed the Practices of More Than 10k Vendors.

Subject Rights Management
Easily Manage 1 or 1,000,000 Data Subject Requests.

Osano Features
Cookie Consent Management, Vendor Risk Monitoring, and More!

<https://www.osano.com> › cookieconsent

Cookie Consent - The most popular solution to cookie laws

The most popular solution to cookie laws. The original free open source cookie consent popup.

More than 100 Billion cookie consents served since 2016.

<https://github.com> › osano › cookieconsent

osano/cookieconsent: A free solution to the EU ... - GitHub

Cookie Consent is a lightweight JavaScript plugin for alerting users about the use of cookies on your website. It is designed to help you quickly comply with ...

Pull requests 41 · README.md · CHANGELOG.md · Actions

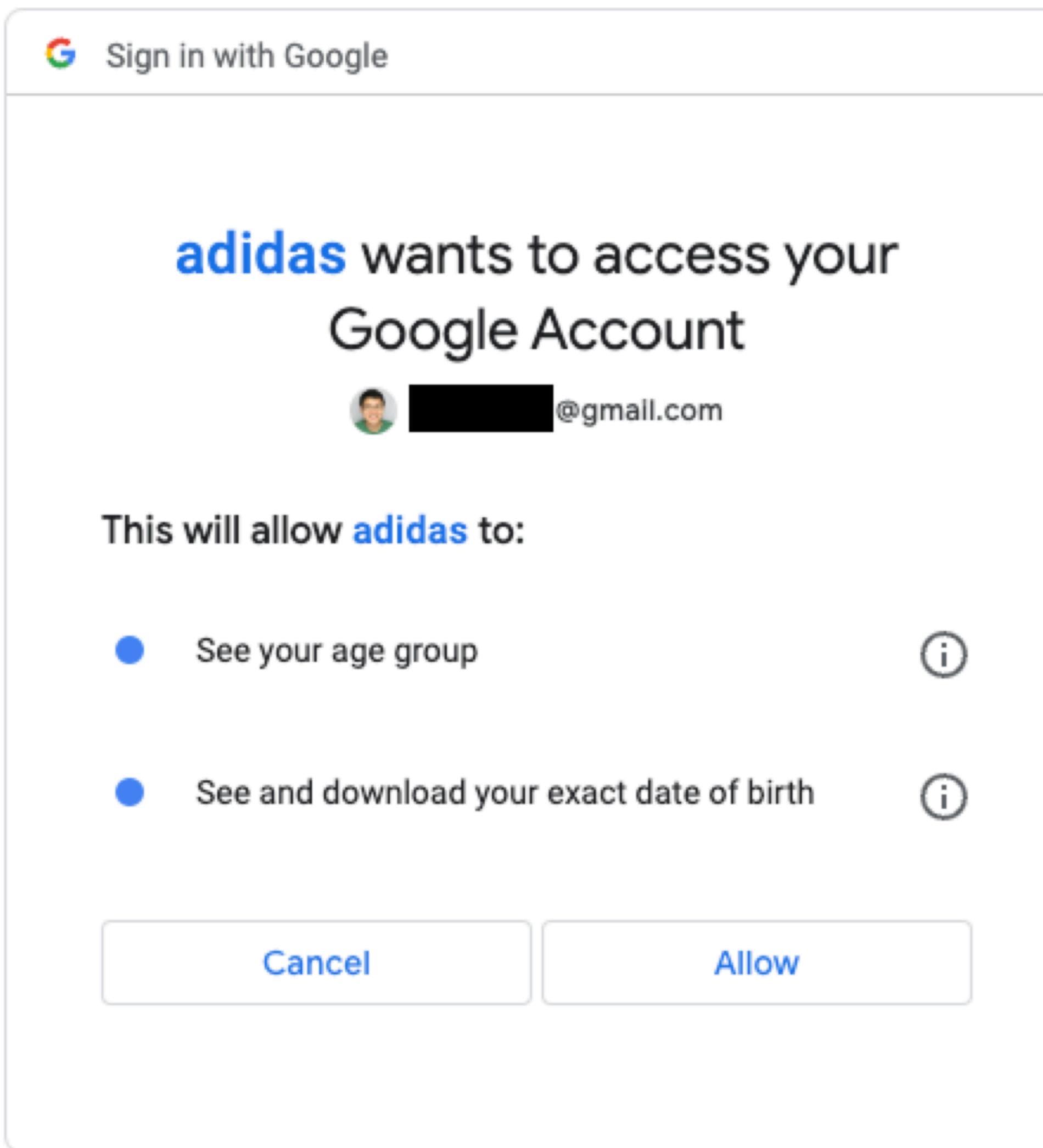
<https://github.com> › orestbida › cookieconsent

- Easy development
- One for all solutions
- Advertisements on high consent rates.

Issues

- Users do not care.
- Developers do not care.
- Advertisers are happy.
- Default opt-in.
- Decentralized management.
- No way to enforce.

OAuth



- How can I let a website access my data?
- Without giving it my password?

Identity use cases (since 2007)

- Simple login - forms and cookies
- Single sign-on across sites
- Mobile app login
- Delegated authorization

Are your friends already on Yelp?

Many of your friends may already be here, now you can find out. Just log in and we'll display all your contacts, and you can select which ones to invite! And don't worry, we don't keep your email password or your friends' addresses. We loathe spam, too.

Your Email Service



Your Email Address

ima.testguy@gmail.com (e.g. bob@gmail.com)

Your Gmail Password

***** (The password you use to log into your Gmail email)

Skip this step

Check Contacts

Step 1
Find Friends

Step 2
Profile Information

Step 3
Profile Picture

Are your friends already on Facebook?

Many of your friends may already be here. Searching your email account is the fastest way to find your friends on Facebook.

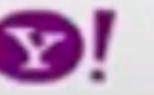
 Gmail

Your Email:

Email Password:

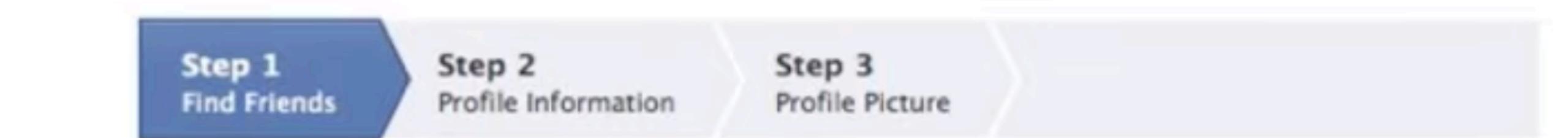
Find Friends

 Facebook will not store your password.

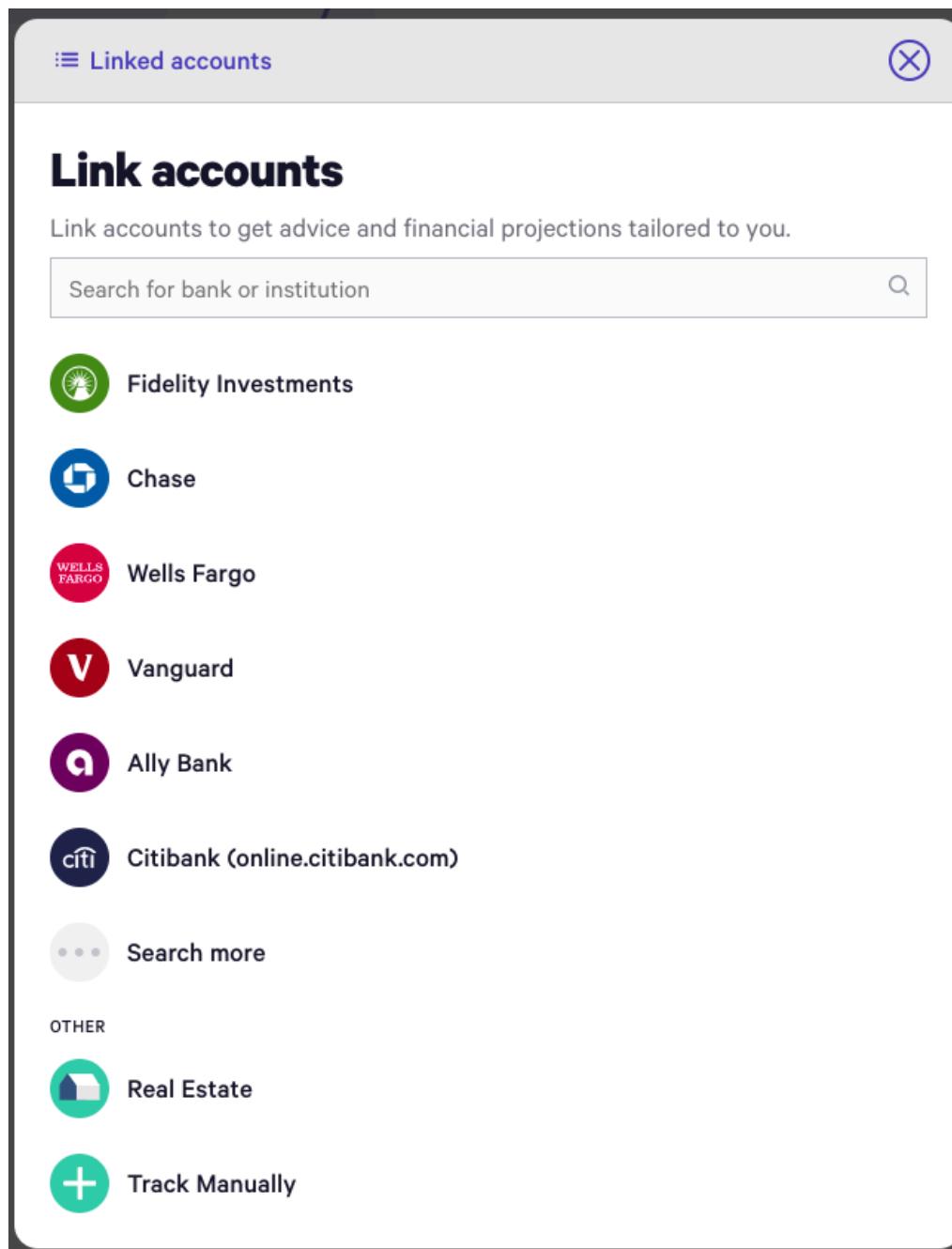
 Yahoo! **Find Friends**

 Windows Live Hotmail **Find Friends**

 Other Email Service **Find Friends**

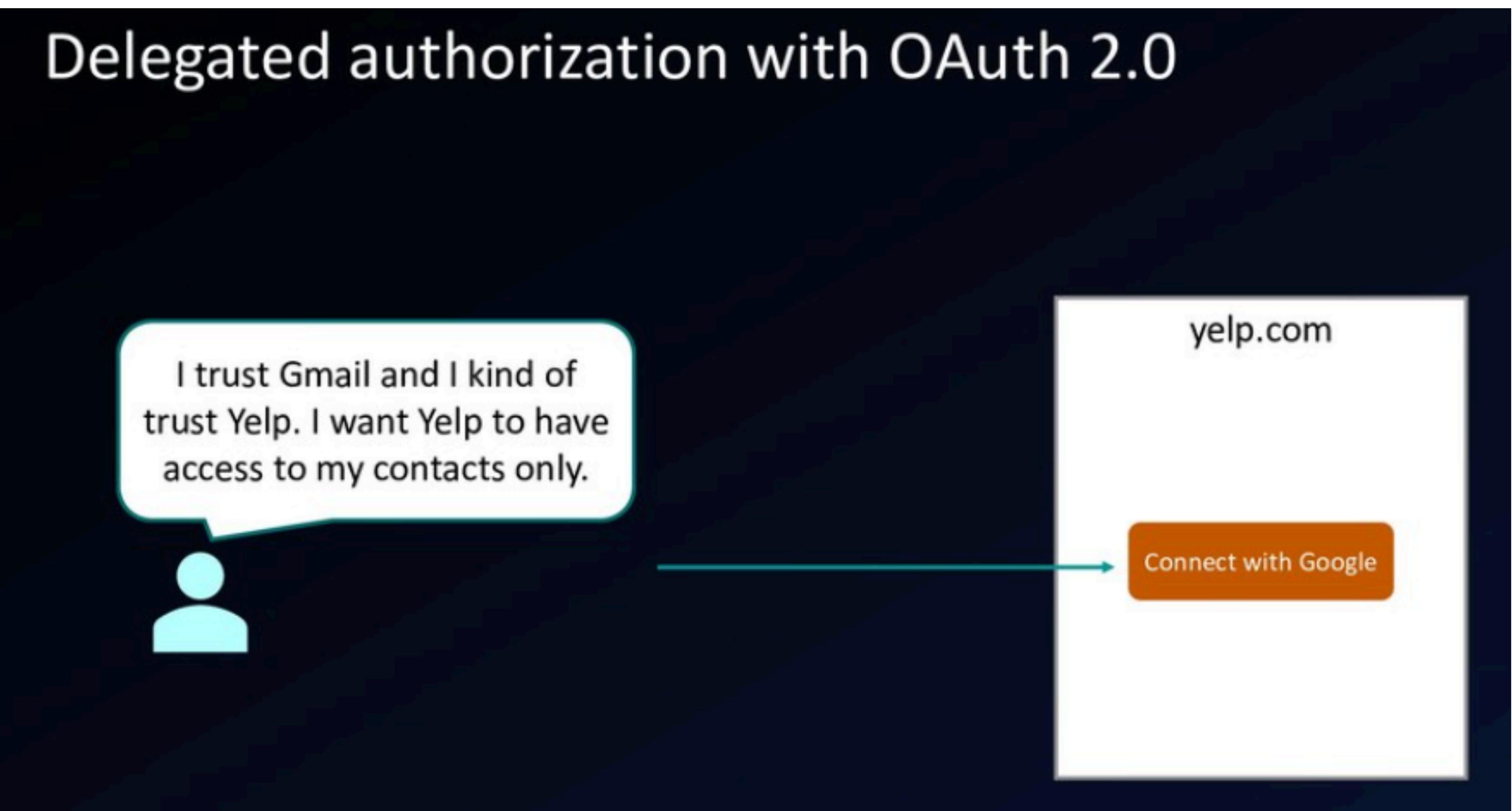


Today!

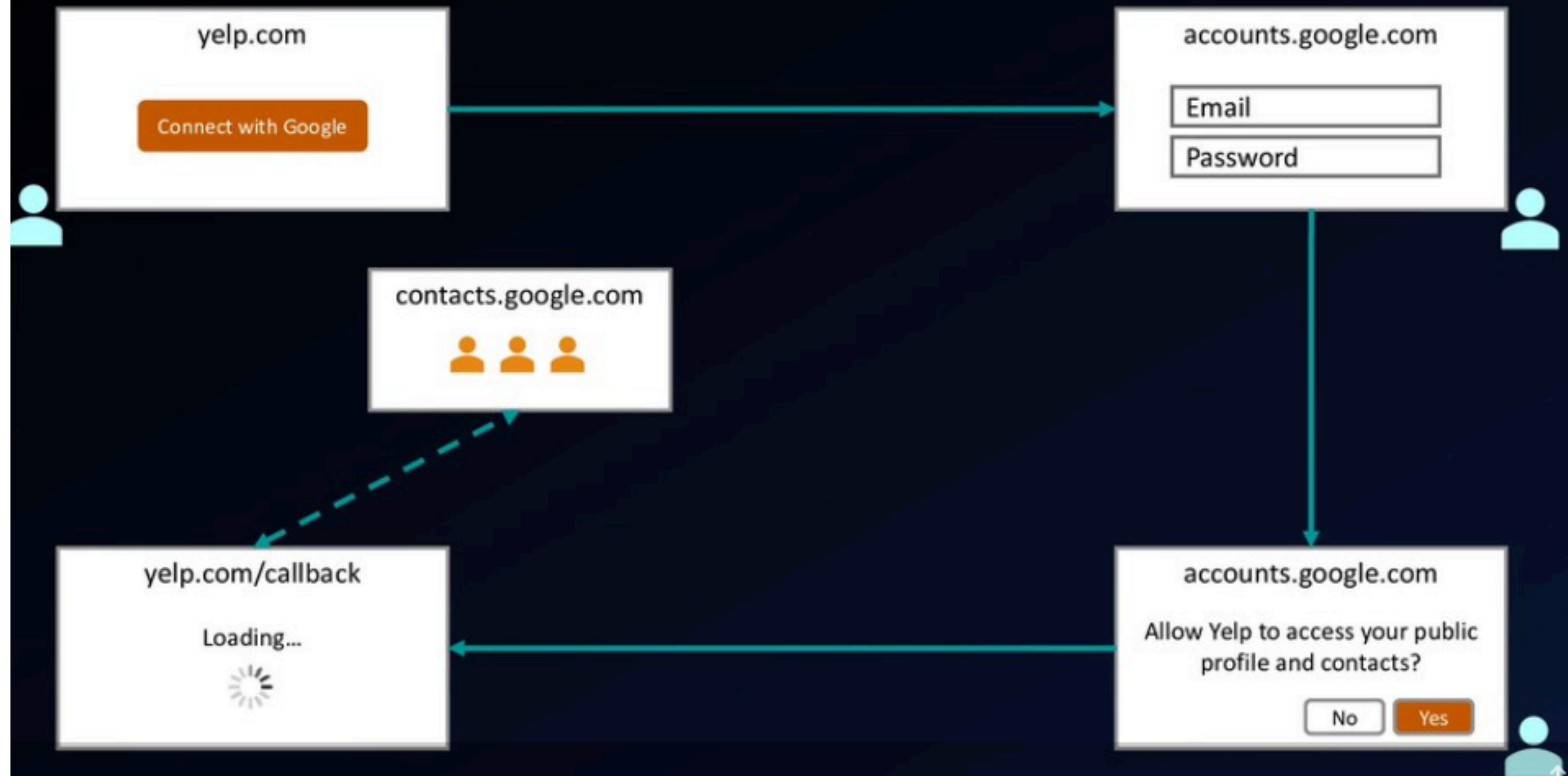


Once you select your institution, you'll be prompted to enter your username and password with that bank or brokerage, and we'll then link to your account. Your security is important to us. We use bank-level security to keep your account safe. Linking does not allow Wealthfront to manage or transfer assets in your linked account. Below are some related FAQs:

Problem formulation



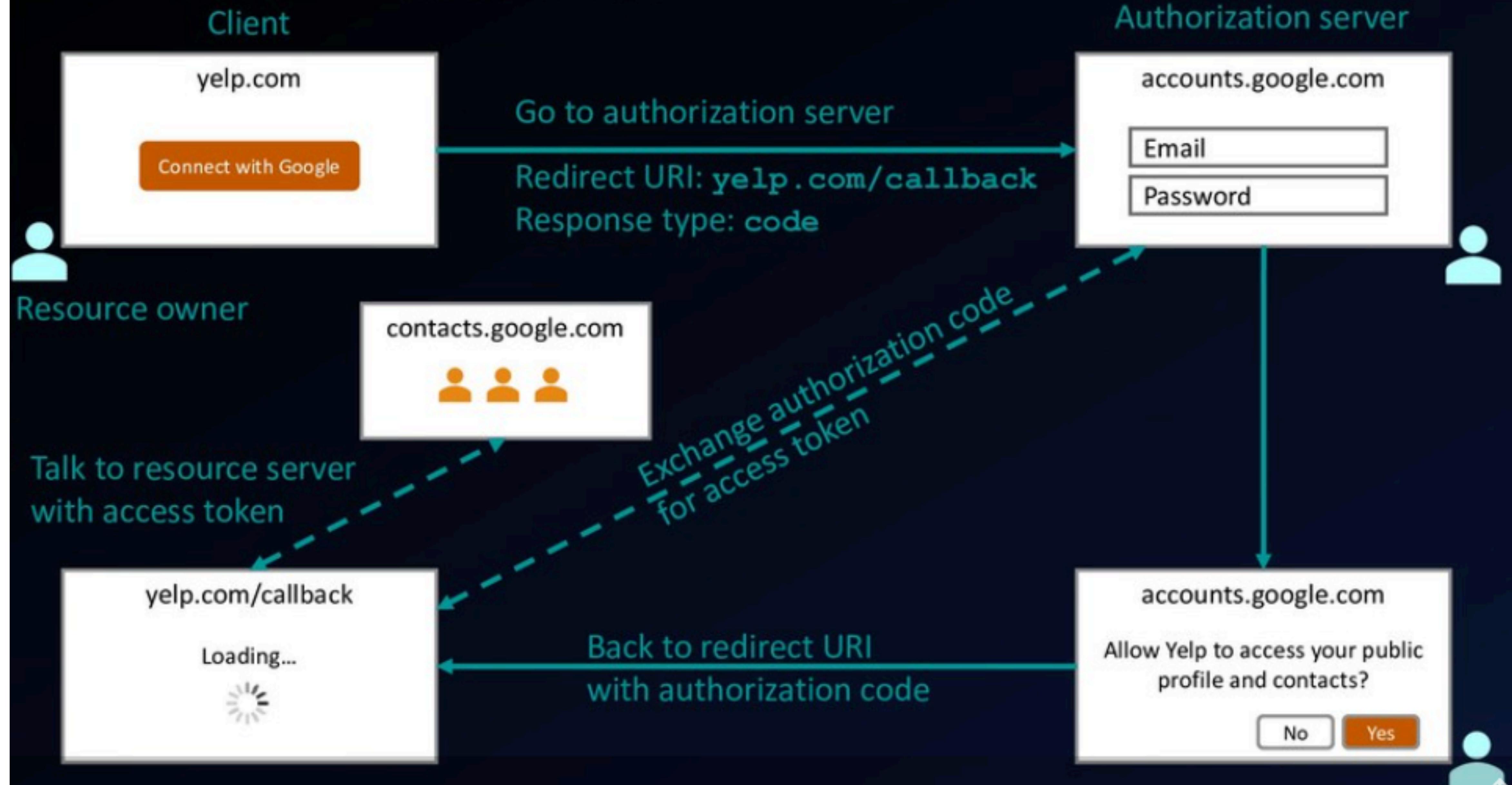
Delegated authorization with OAuth 2.0



OAuth 2.0 Terminology

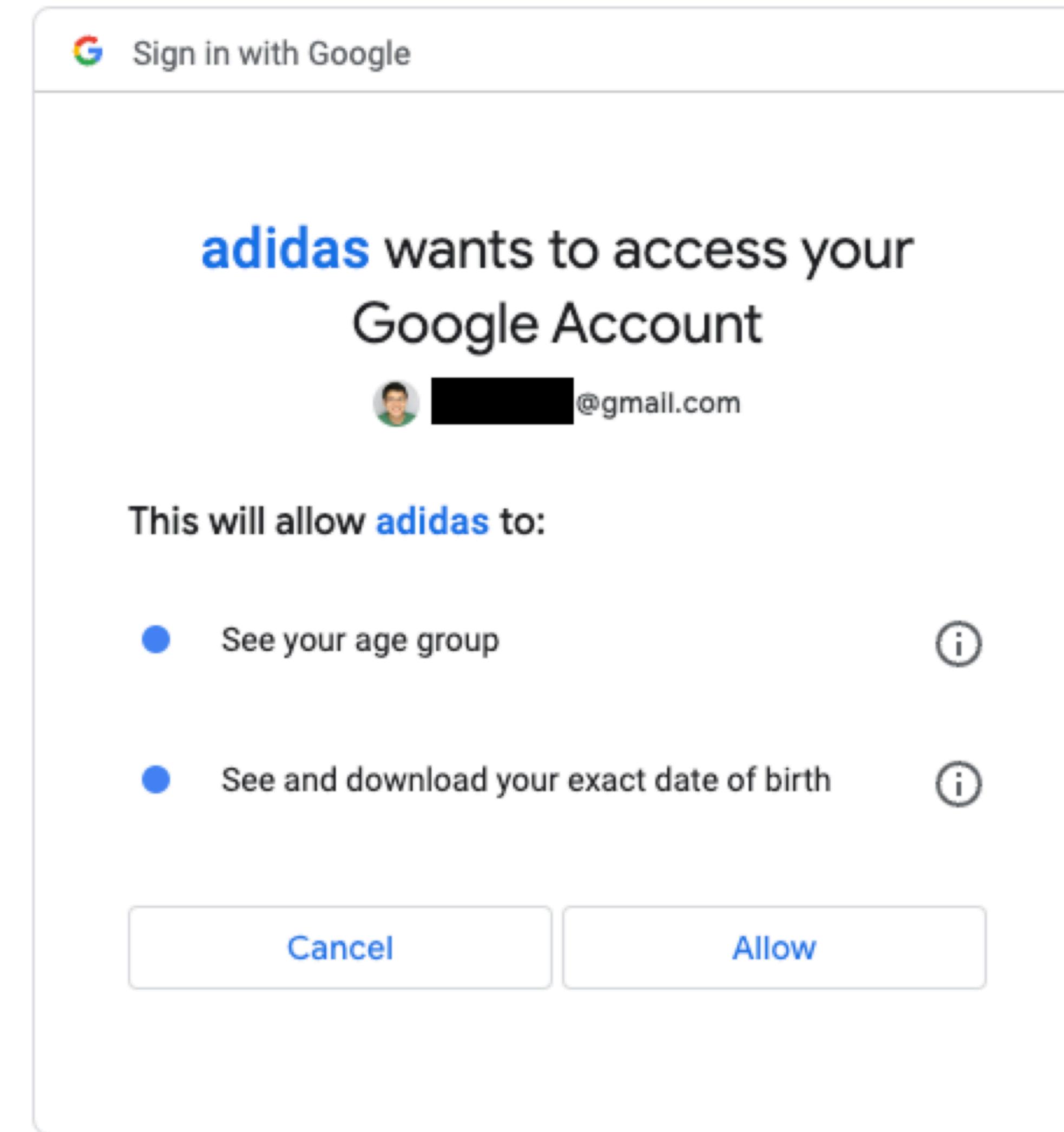
- Resource owner
- Client
- Authorization server
- Resource server
- Authorization grant
- Access token

OAuth 2.0 authorization code flow

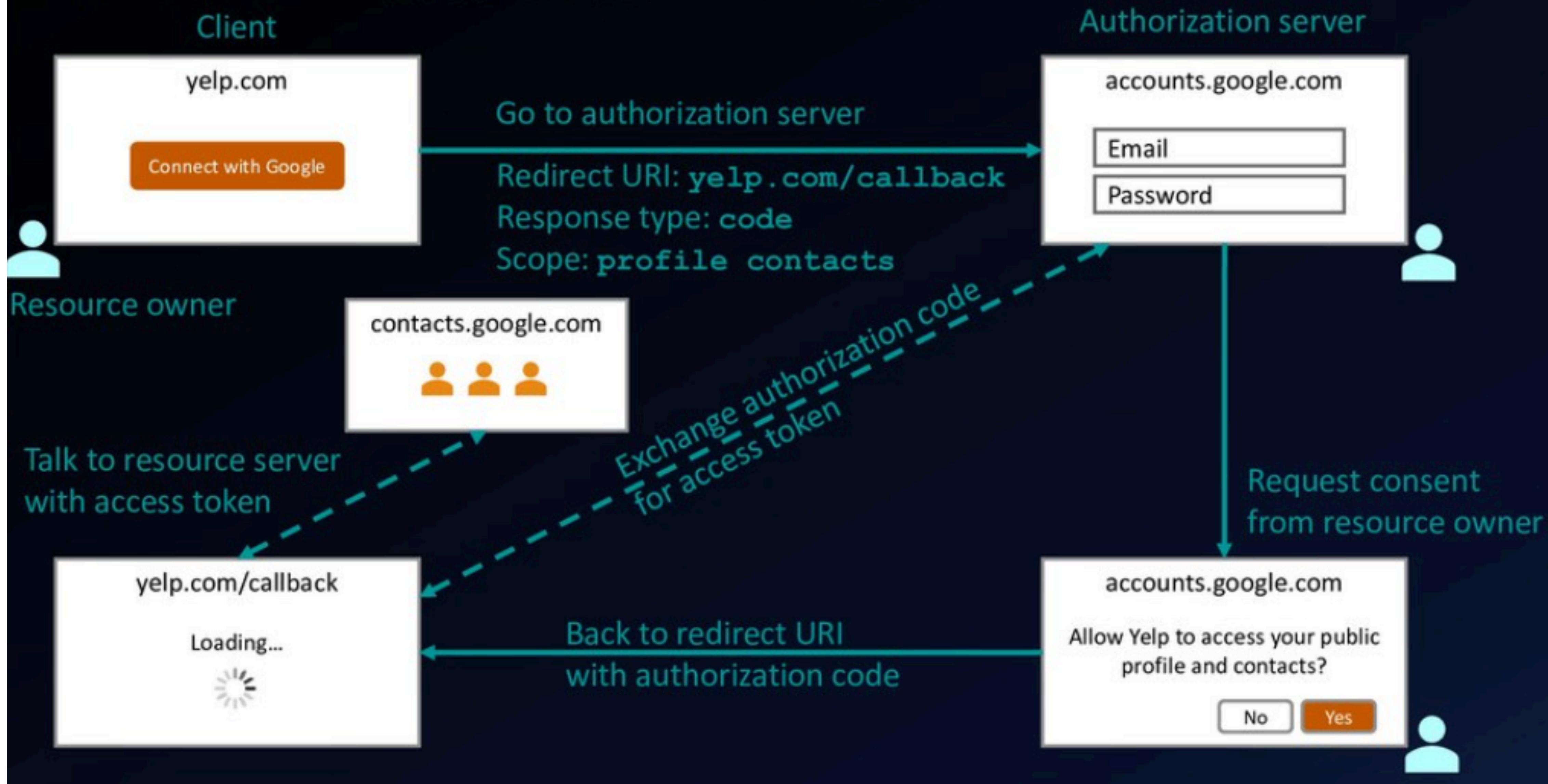


More Terminology

- Scope
- Consent
- Front channel
- Back channel



OAuth 2.0 authorization code flow



Starting the flow

```
https://accounts.google.com/o/oauth2/v2/auth?  
client_id=abc123&  
redirect_uri=https://yelp.com/callback&  
scope=profile&  
response_type=code&  
state=foobar
```

Calling back

```
https://yelp.com/callback?  
error=access_denied&  
error_description=The user did not consent.
```

```
https://yelp.com/callback?  
code=oMsCeLvIaQm6bTrgtp7&  
state=foobar
```

Exchange code for an access token

```
POST www.googleapis.com/oauth2/v4/token
```

```
Content-Type: application/x-www-form-urlencoded
```

```
code=oMsCeLvIaQm6bTrgtp7 &
```

```
client_id=abc123 &
```

```
client_secret=secret123 &
```

```
grant_type=authorization_code
```

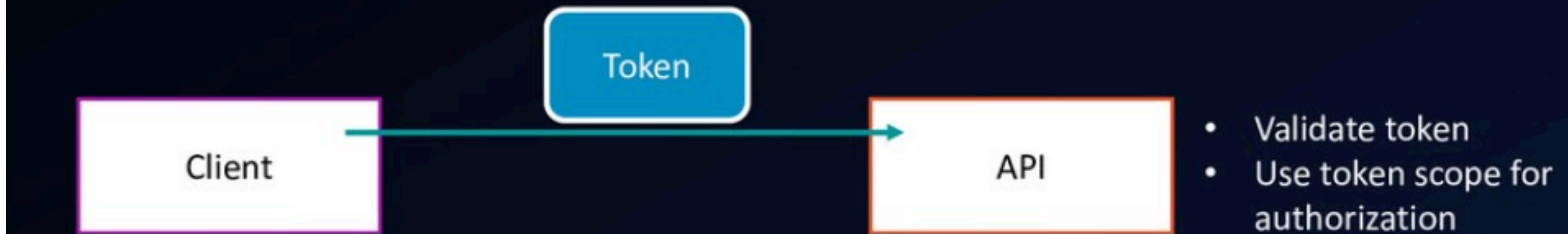
Authorization server returns an access token

```
{  
    "access_token": "fFAGRNJrulFTz70BzhT3Zg",  
    "expires_in": 3920,  
    "token_type": "Bearer",  
}
```

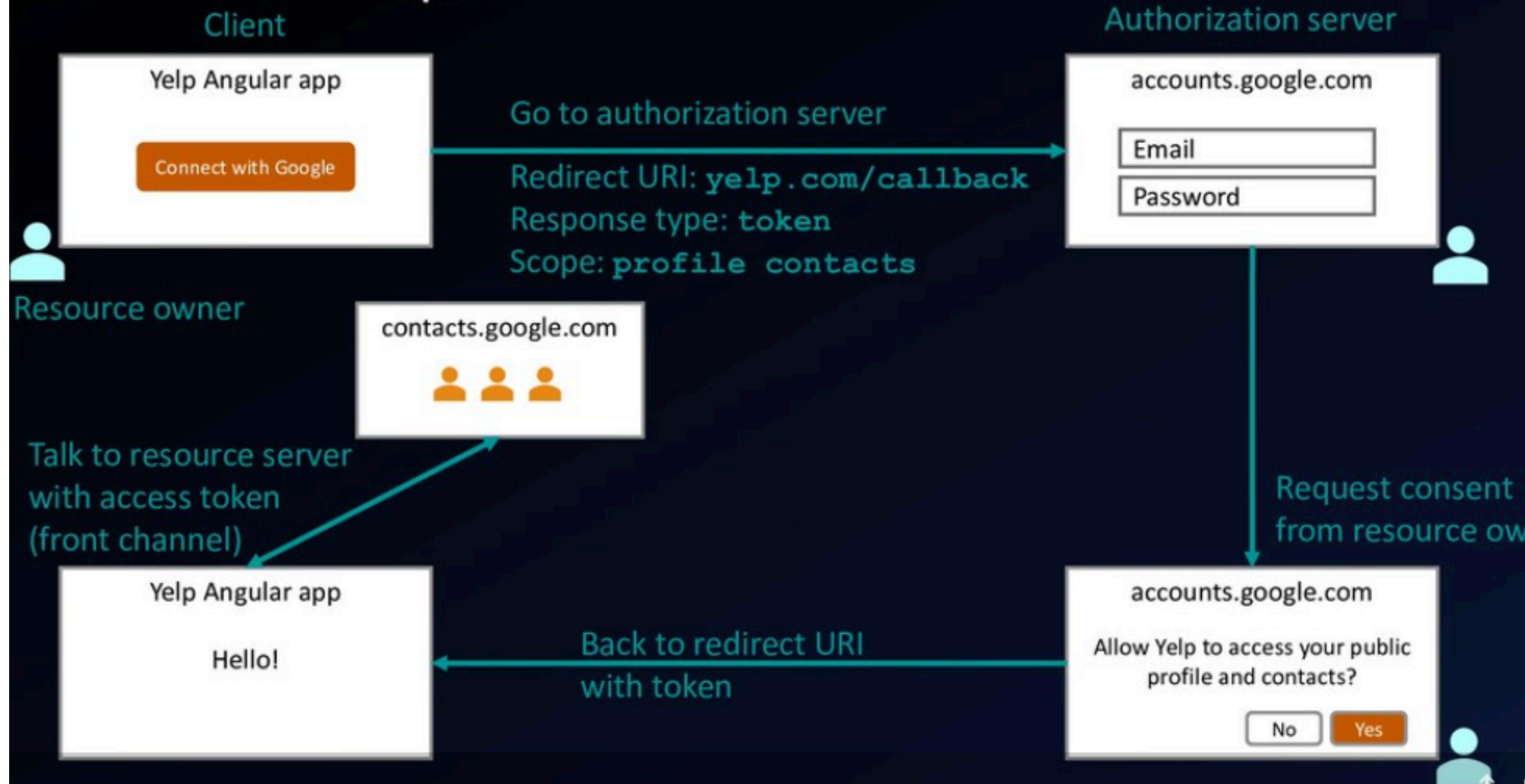
Use the access token

```
GET api.google.com/some/endpoint
```

```
Authorization: Bearer fFAGRNJru1FTz70BzhT3Zg
```



OAuth 2.0 implicit flow

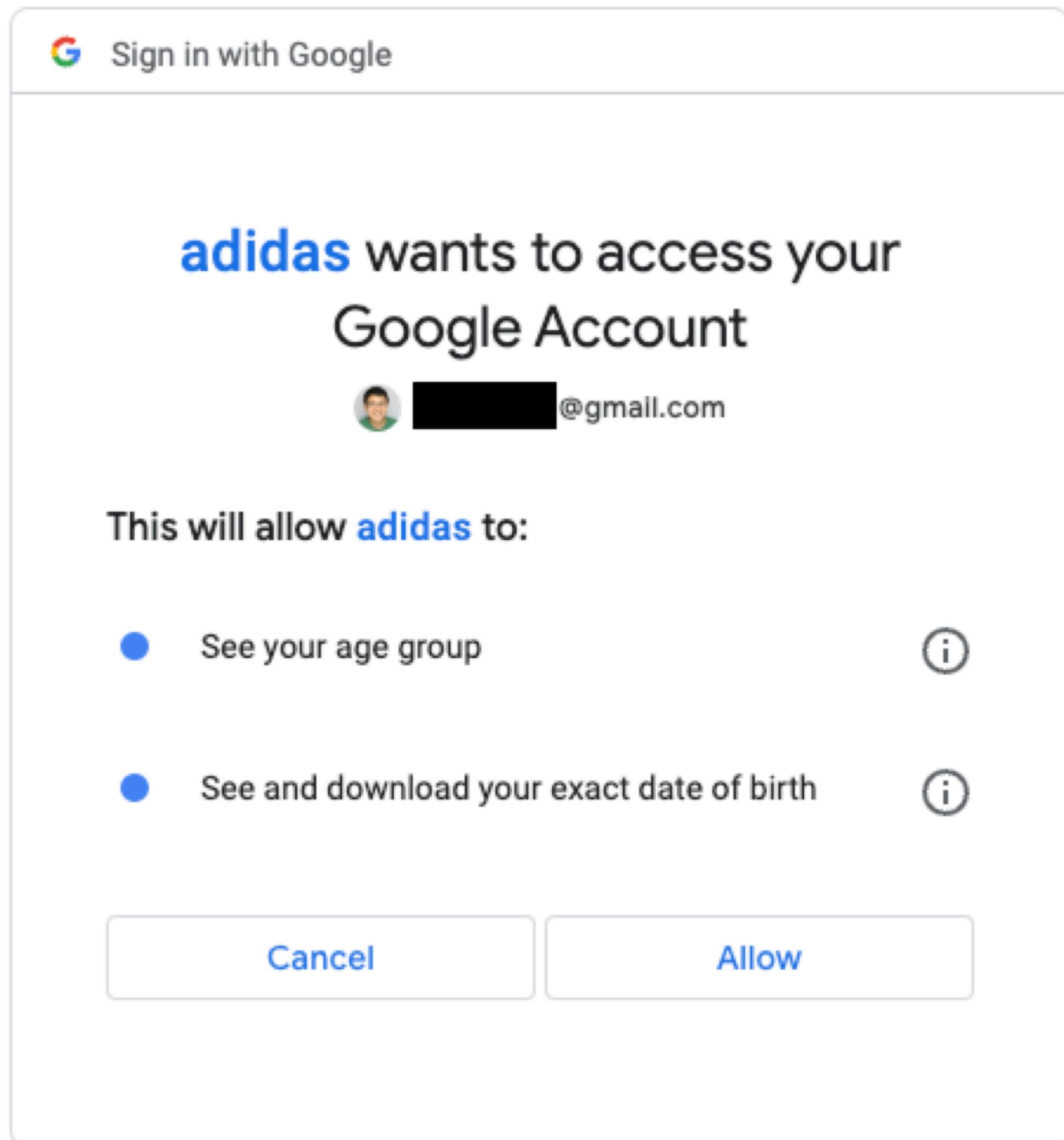


Identity use cases (since 2007)

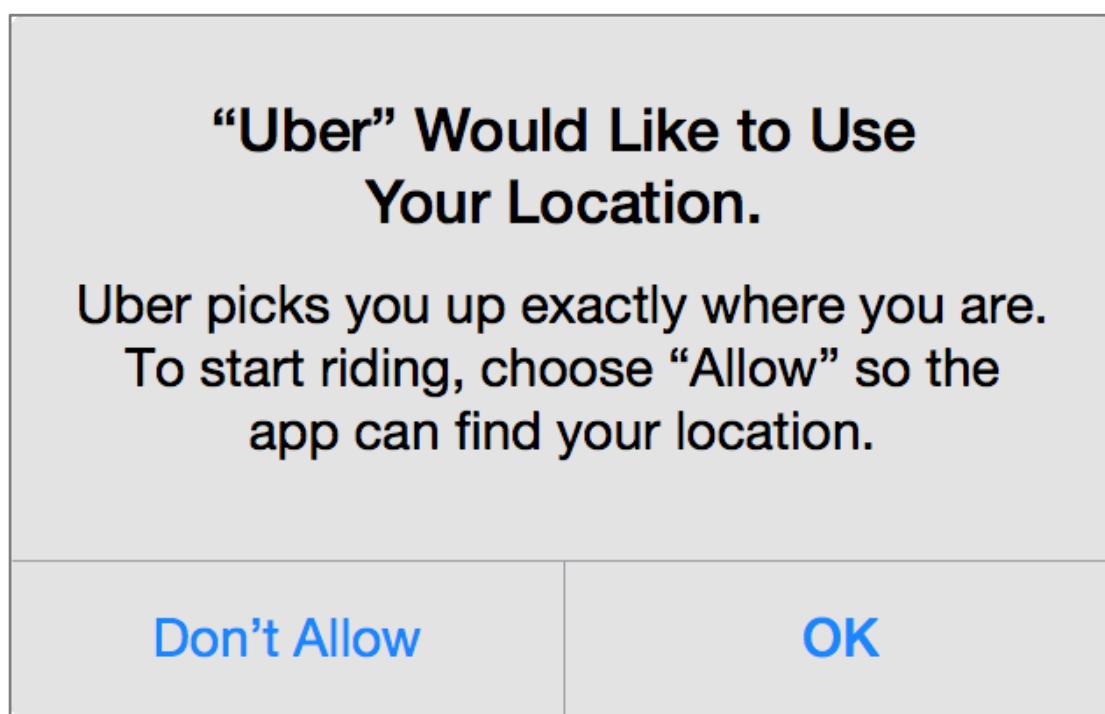
- Simple login - forms and cookies - Authentication
- Single sign-on across sites - Authentication
- Mobile app login - Authentication
- Delegated authorization - Authorization

Problems with OAuth

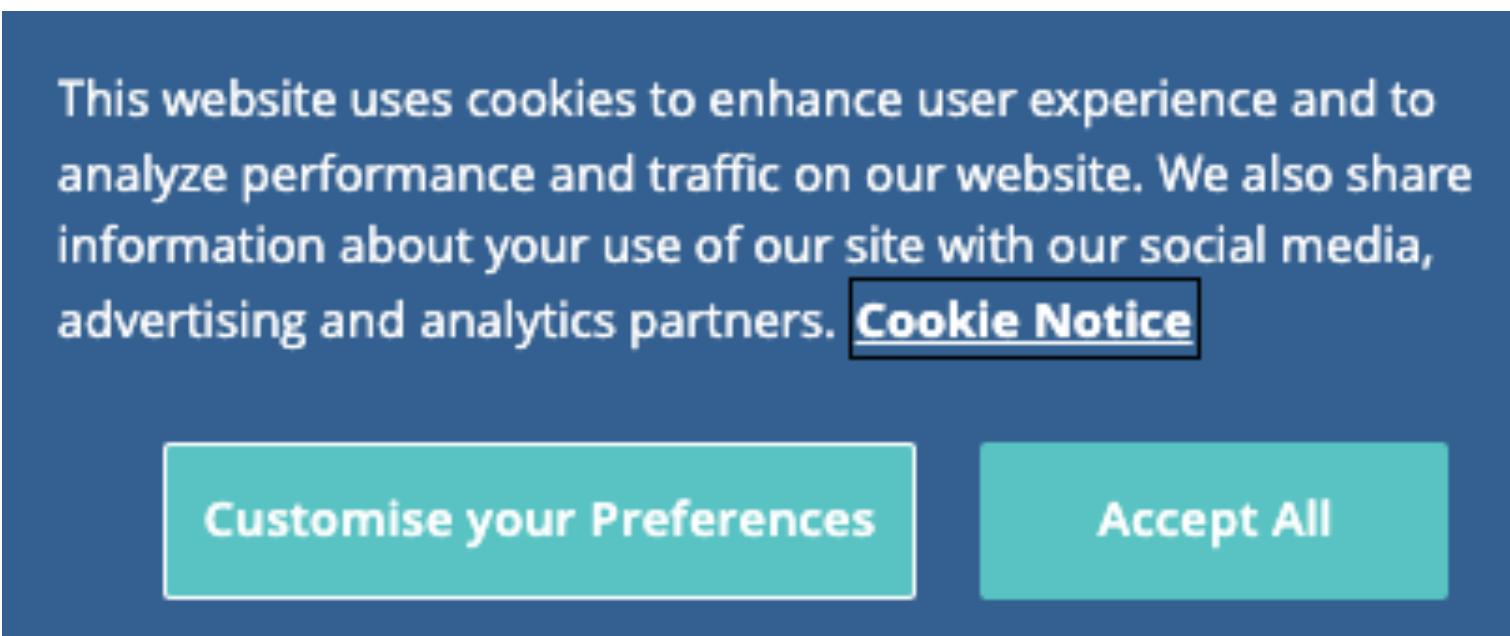
- No standard way to get the user's information
- Every implementation is a little different
- No common set of scopes
- Coarse granularity
- Users do not pay attention
- Hard to understand the terms



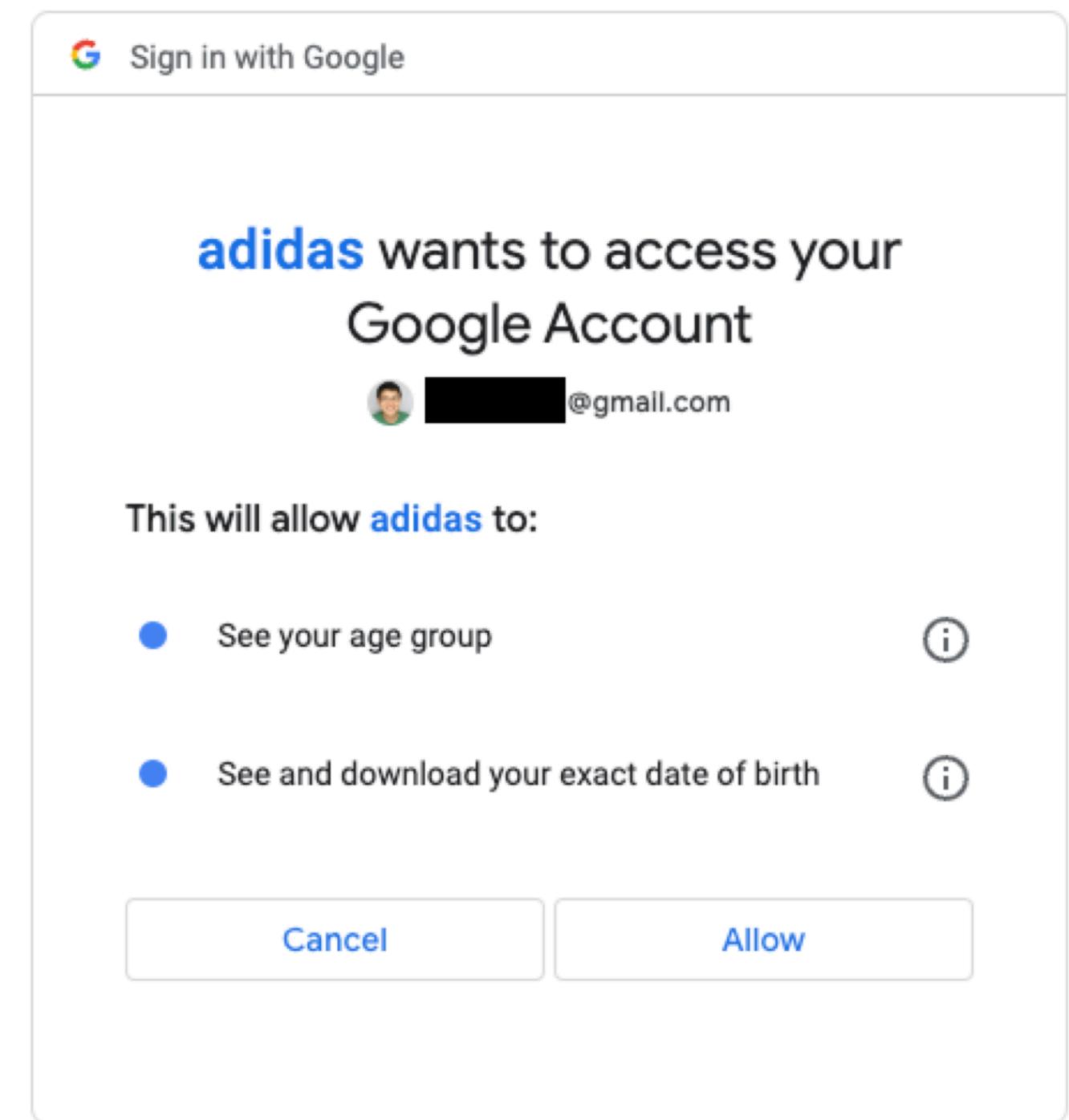
Recap: Three systems



Android permission

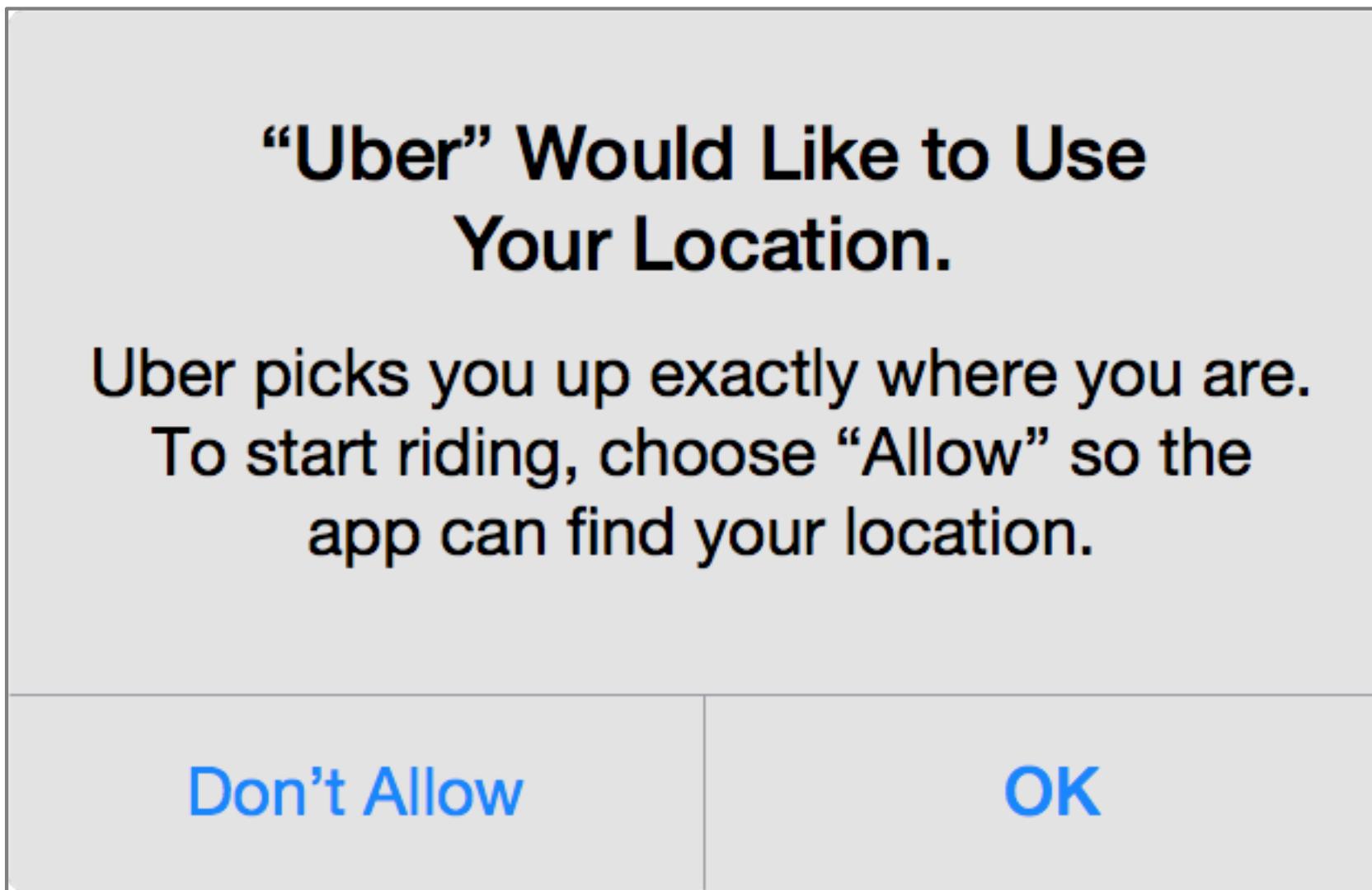


Browser cookie consent



OAuth

Recap: Key privacy concepts



- Control
- Notice
- Consent
- Usability
- System mechanisms