



# ClientLogin #FAIL

Dirk Balfanz  
5/10/2011

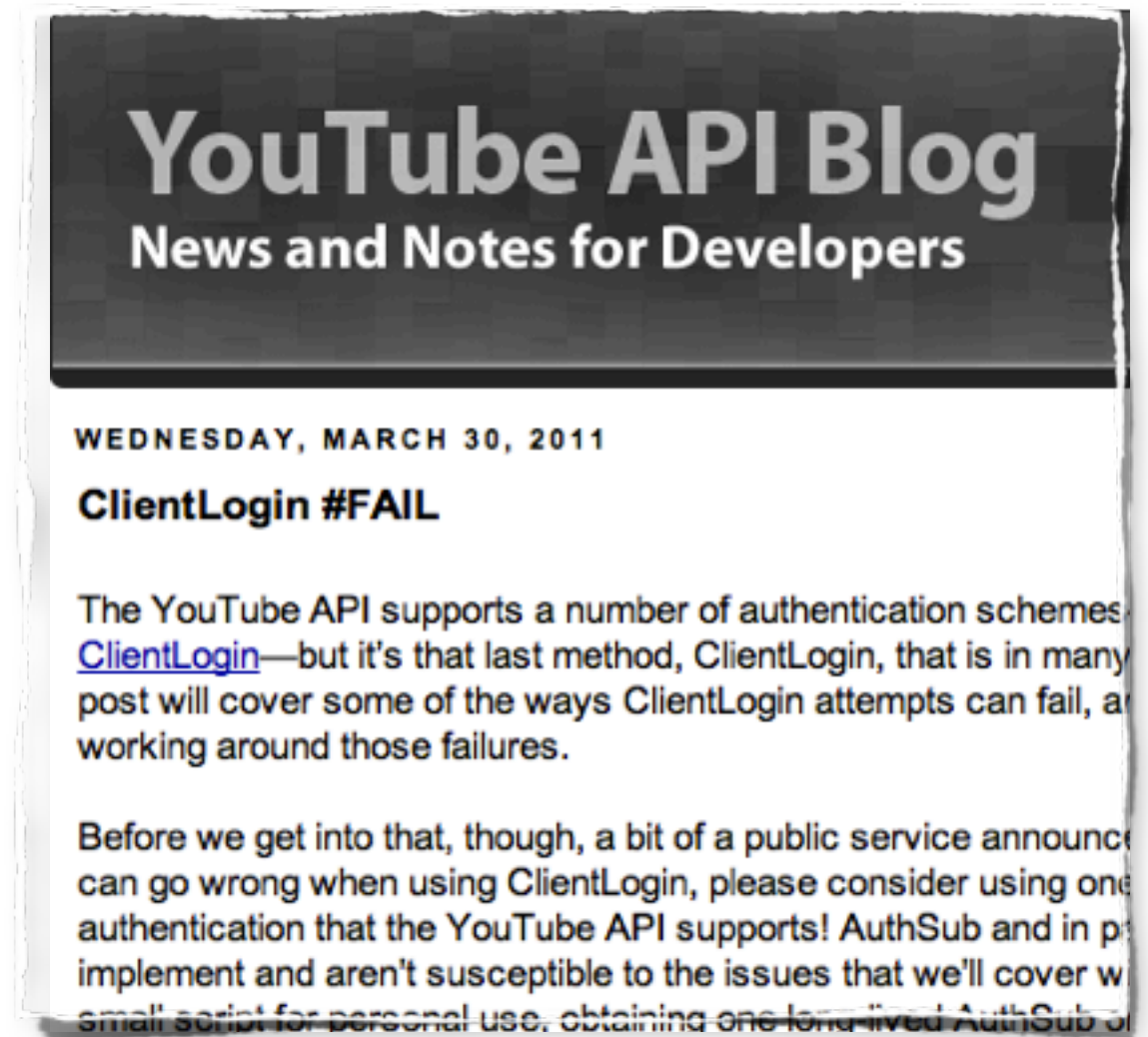


**feedback** <http://goo.gl/2Qxx6>

**hashtags** #io2011 #GoogleAPIs

# Agenda

- Installed applications and ClientLogin
- Why ClientLogin is not a good choice for your users
  - ClientLogin breaks for many different use cases
  - see [previous blog posts](#)
- Application-Specific Passwords are a stop-gap
  - fix some, but not all, of the above use cases
- OAuth 2.0 is the Official Solution
  - best practices for moving from ClientLogin to OAuth

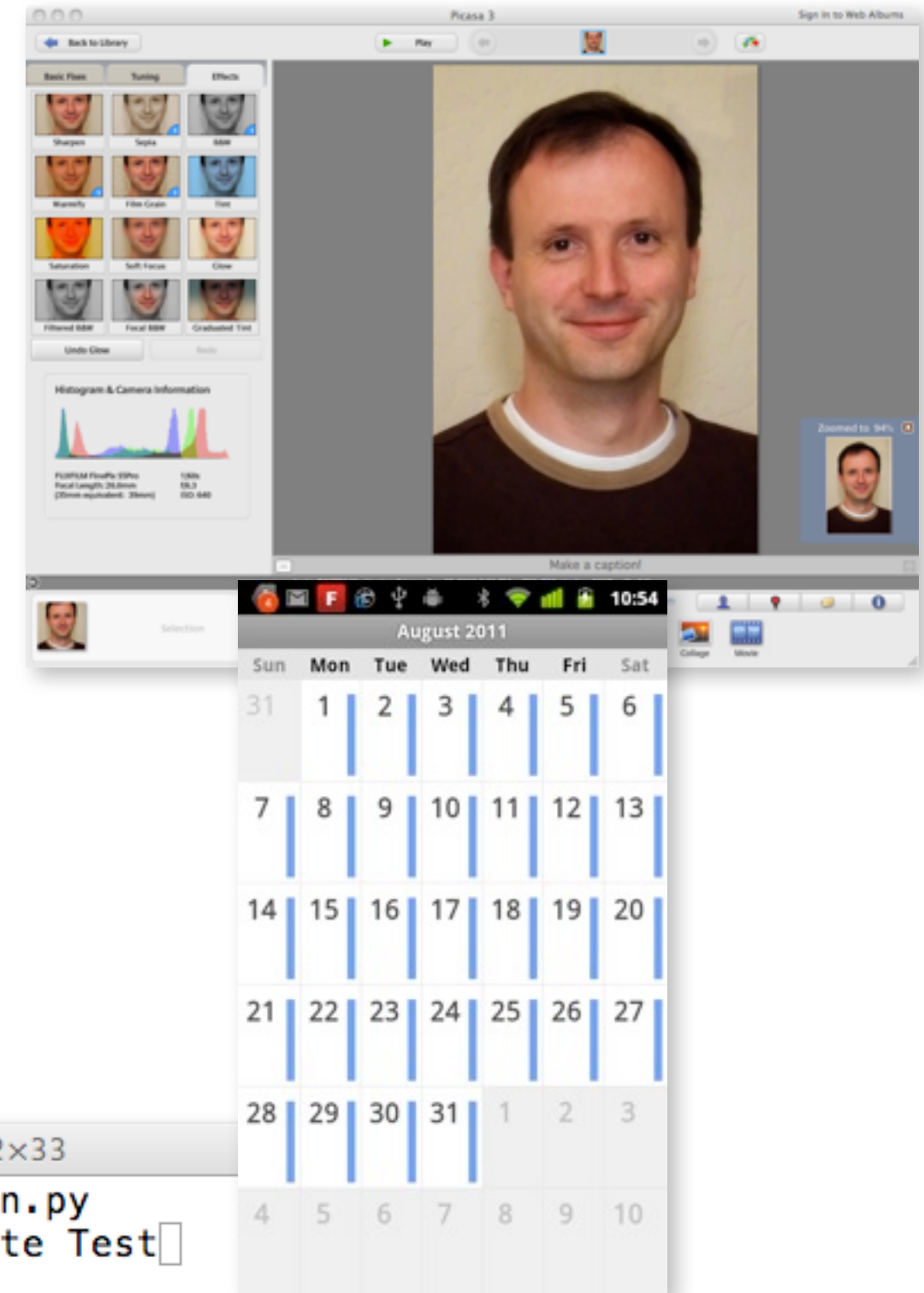


# What Are Installed Apps?

*Anything the user installed on their device that is **not bound** by the rules of the web*

Examples:

- Video editor that uploads to YouTube
- Picasa photo viewer on TiVo
- IMAP and XMPP clients
- Calendar integration tool on iPhone or Android
- Command line tools



# How It Works Today: ClientLogin



Applications ask for a user's email address and password

The screenshot shows the Google Talk application window. At the top, it says "Google Talk" with a "Settings | Help" link. Below the logo, there is a "Sign in" button and the status "Offline". The sign-in form includes a "Username:" label and an empty text input field, a "Password:" label and an empty password input field, and a checked checkbox for "Remember password". A "Sign In" button is located below the form. At the bottom, there are links for "Forgot your password?" and "Don't have an account?".

The screenshot shows a "Picasa™ Web Albums" sign-in dialog box overlaid on a photo gallery. The dialog has a title "Sign in to Web Albums" and a brief description: "With Web Albums, you can share online photo albums with friends and family, or create public albums to share with the world. It's free, and easy to use." Below this is a link "Click here to learn more.". The main section is titled "Sign in to Web Albums with your Google Account" and contains a "Username:" label and an empty text input field, a "Password:" label and an empty password input field, and a checked checkbox for "Remember me on this computer". There are "Cancel" and "Sign In" buttons. At the bottom, there are links for "Forgot your Password?" and "Sign up for Web Albums".

# When ClientLogin Fails...



# ClientLogin FAIL: CAPTCHA

Sign in with your  
**Google Account**

Username:   
ex: pat@example.com

Password:

Enter the correct password above and then type  
the characters you see in the picture below.





Enter the letters as they are shown in the image  
above.  
*Letters are not case-sensitive*

Stay signed in

[Can't access your account?](#)



# ClientLogin FAIL: 2-Step Verification

1.

Sign in with your  
**Google Account**

Email:   
ex: pat@example.com

Password:

Stay signed in

[Can't access your account?](#)



2.

**Google accounts**

**Enter verification code**

To verify your identity on this computer, enter the verification code generated by your mobile application.

Enter code:

Remember verification for this computer for 30 days.

[Other ways to get a verification code »](#)



# ClientLogin FAIL: OpenID

Sign in with your  
**Google Account**

Email:   
ex: pat@example.com

Password:

To sign in with this account, click Continue.  
[About signing in](#)

Stay signed in

[Can't access your account?](#)

## Google accounts

### How to sign in with your OpenID

If you use your OpenID to access your Google Account, then Google doesn't know the password associated with it.

To sign in with your OpenID from the Google Accounts page:

Type your full email address

Leave the password field blank

Click "Sign in"

You will be directed to the identity provider's sign-in page, where you can enter your password.



# ClientLogin FAIL: Account Wizards

The screenshot shows the YouTube account linking wizard. At the top, there is the YouTube logo, a search bar, and navigation links for 'Search', 'Browse', 'Upload', and 'Sign Out'. The main heading is 'Update your YouTube account by linking to a Google Account'. Below this, a message states: 'You will no longer be able to sign in to YouTube without a Google Account. If you do not want to link this YouTube account, you can [sign out here](#).' Under the heading 'Why update?', there is a section for 'Improved security' with a padlock icon and the text 'Better security reduces the chance of account theft.' To the right, a box displays the current YouTube account information: 'Username: sachsvideos'. A large right-pointing arrow labeled 'Link to:' connects this box to the account linking form. The form, titled 'Already have an account? Sign in here', contains the following fields: 'Your current email address:' with a text input and a note 'e.g. myname@example.com. This will be used to sign-in to your account.'; 'Choose a password:' with a text input and a note 'Minimum of 8 characters in length. Password strength:'; 'Re-enter password:' with a text input; and 'Word Verification:' with a visual captcha showing the word 'Darnight' in red cursive and a text input field with an accessibility icon. At the bottom of the form is a blue button labeled 'Create and link accounts'.

# ClientLogin FAIL: Other



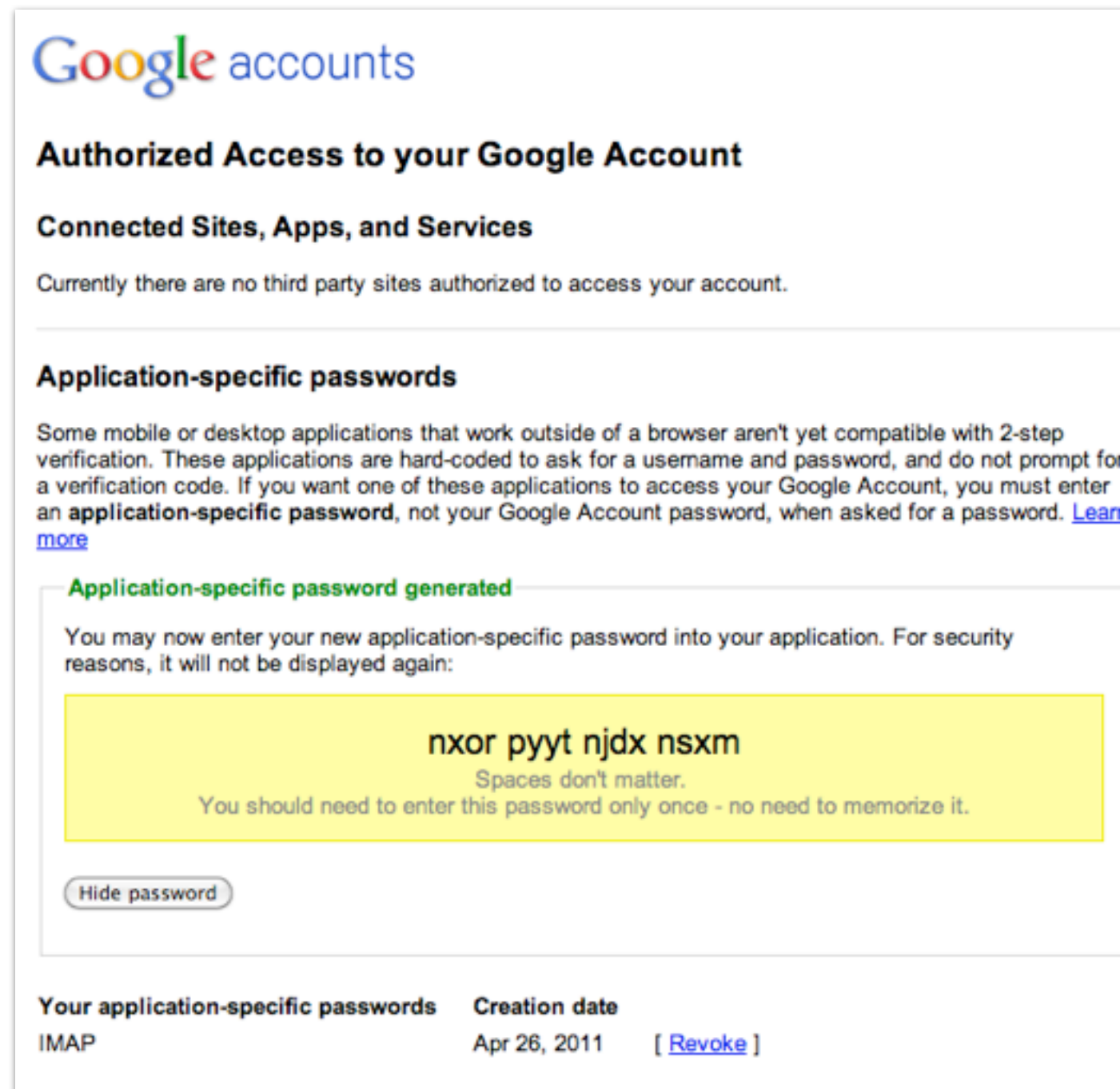
- Google Apps for schools and enterprises using SAML
- Other account wizards (e.g., hijacked accounts)
- Future potential speed bumps during authentication

When ClientLogin Fails...

...Use Application-Specific Passwords!

# Stop-Gap: Application-Specific Passwords

- Just like passwords, but
  - you can have many
  - you can revoke them
- Useful for:
  - IMAP
  - XMPP
  - Android
  - Exchange clients
  - ...



The screenshot shows the Google accounts interface for managing application-specific passwords. It includes the Google logo, a heading for 'Authorized Access to your Google Account', and a section for 'Application-specific passwords'. A yellow box displays a generated password: 'nxor pyyt njdx nsxm', with a note that spaces don't matter and it should be used only once. Below the password is a 'Hide password' button. At the bottom, a table lists the application-specific passwords.

**Google** accounts

**Authorized Access to your Google Account**

**Connected Sites, Apps, and Services**

Currently there are no third party sites authorized to access your account.

**Application-specific passwords**

Some mobile or desktop applications that work outside of a browser aren't yet compatible with 2-step verification. These applications are hard-coded to ask for a username and password, and do not prompt for a verification code. If you want one of these applications to access your Google Account, you must enter an **application-specific password**, not your Google Account password, when asked for a password. [Learn more](#)

**Application-specific password generated**

You may now enter your new application-specific password into your application. For security reasons, it will not be displayed again:

**nxor pyyt njdx nsxm**  
Spaces don't matter.  
You should need to enter this password only once - no need to memorize it.

[Hide password](#)

Your application-specific passwords	Creation date	
IMAP	Apr 26, 2011	[ <a href="#">Revoke</a> ]

# Stop-Gap: Application-Specific Passwords

Pros	Cons
Works with legacy clients	Bad user experience
Works with different user accounts (SAML, 2factor, etc.)	Some account types still don't work (unlinked YouTube, suspended, etc.)



When ClientLogin Fails...

~~...Use Application-Specific Passwords!~~  
...Stop Using ClientLogin!!!

Isn't OAuth just  
for Web Apps?

No!  
OAuth is available  
for installed apps  
as well!

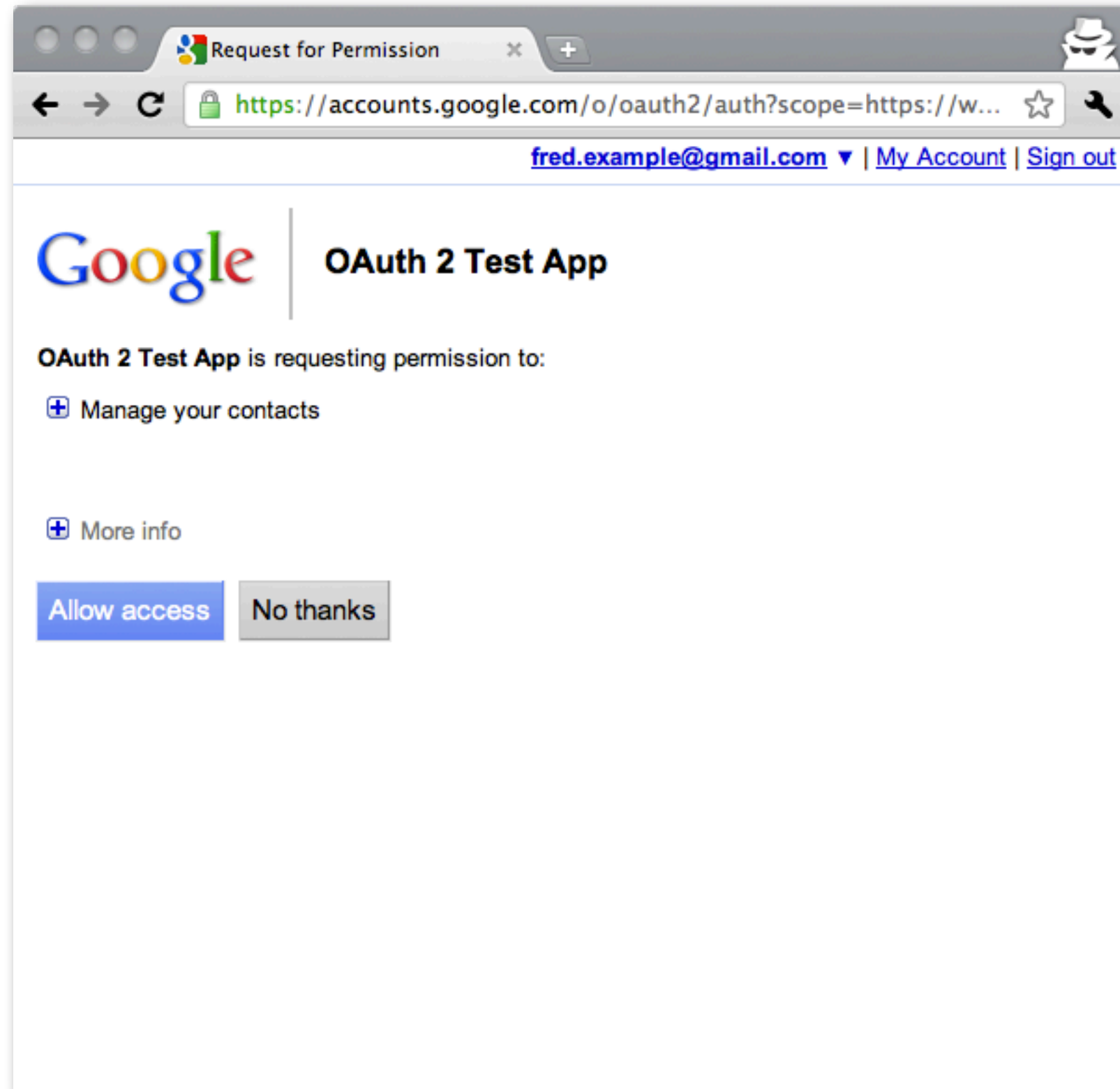
You don't even  
need a browser!  
(more later...)

# Solution: OAuth 2.0 Standard

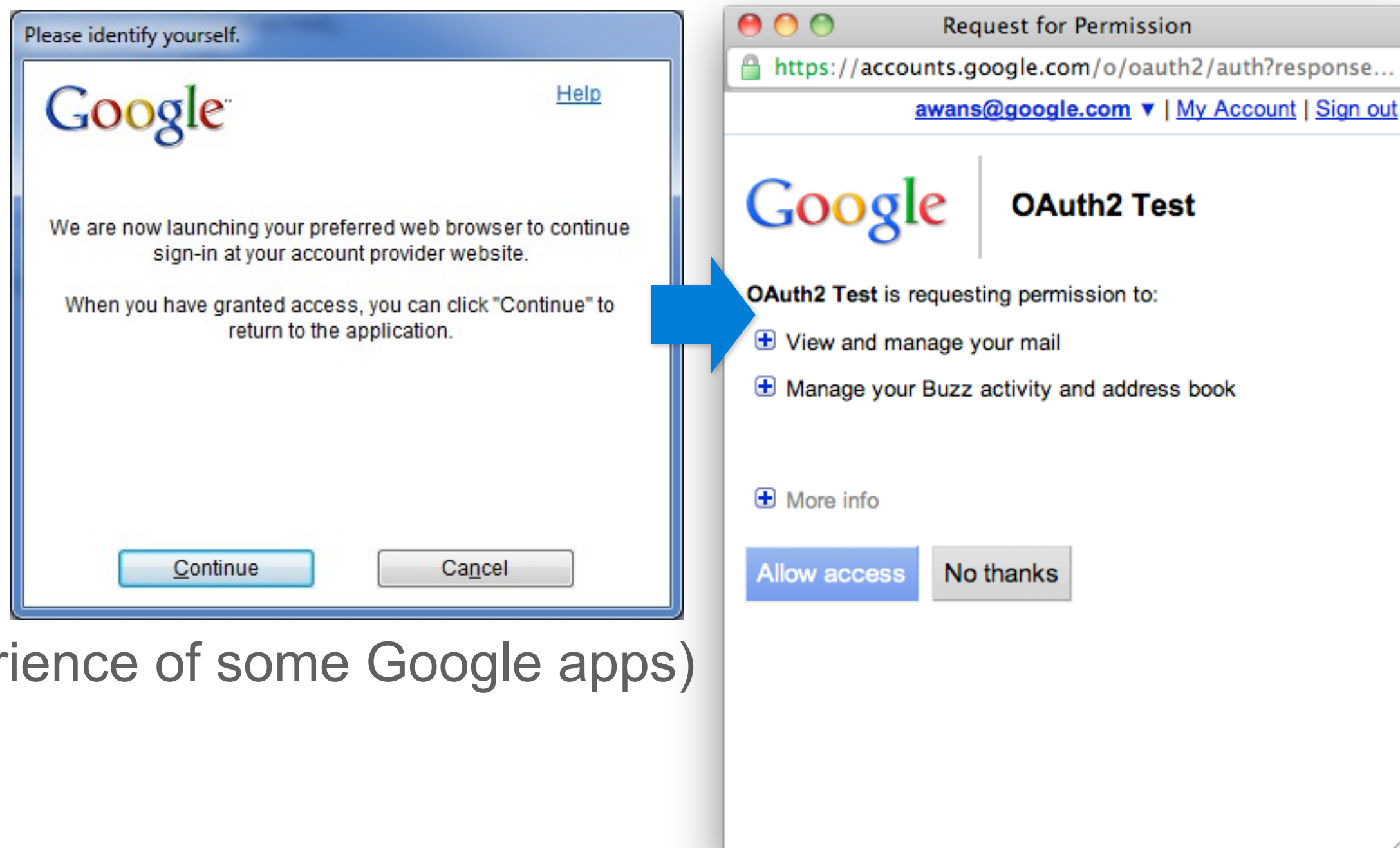
- General user experience:
  - Installed app opens a web-browser (on same or different device)
  - User authenticates
  - User grants access
- Key technical details
  - Installed app stores a long-lived refresh token
  - API calls are made by including a short-lived access token that you can request using the refresh token
  - For details, search for "OAuth 2.0 Google"



# Reminder/Comparison: OAuth on a Web App



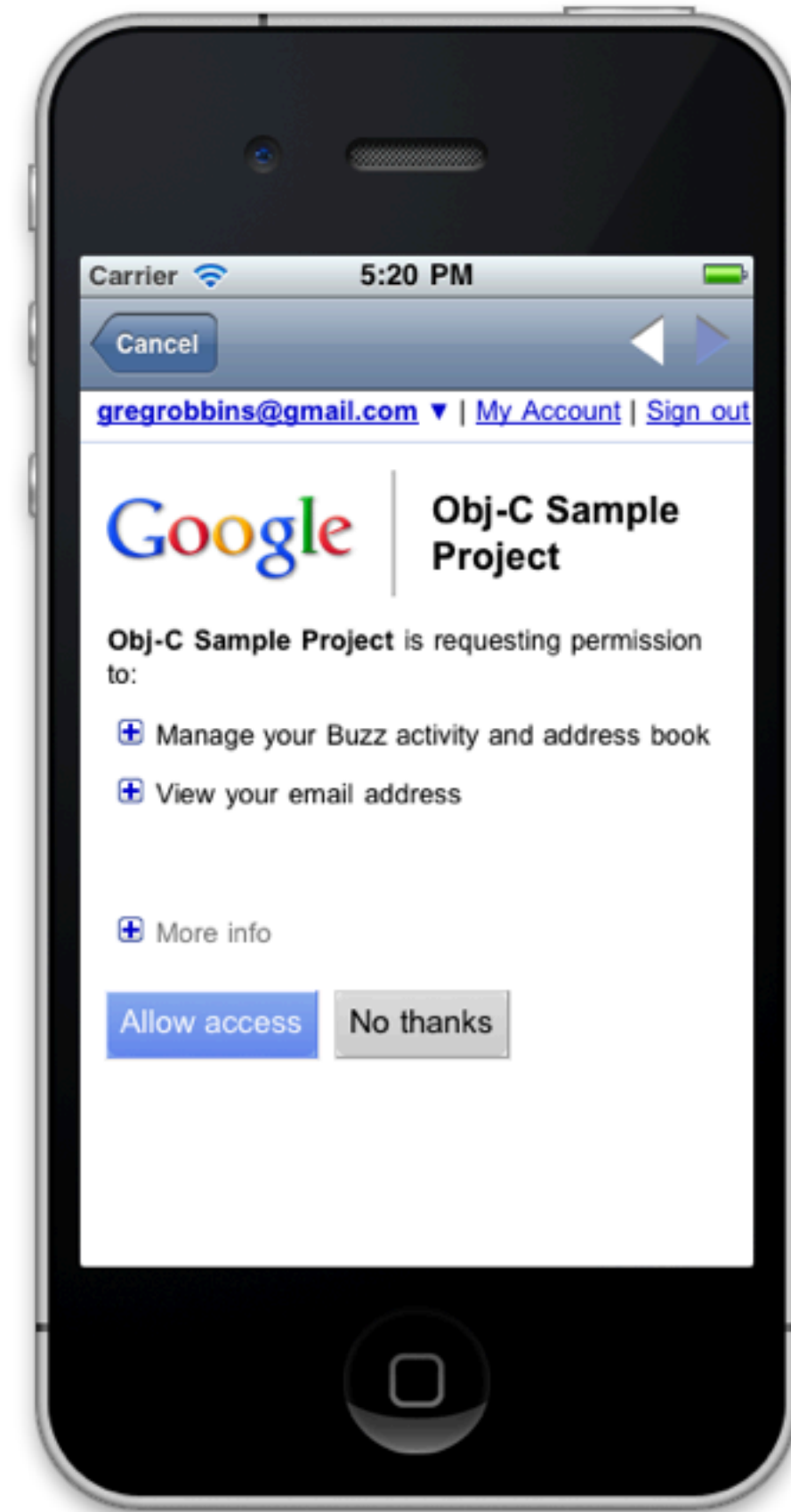
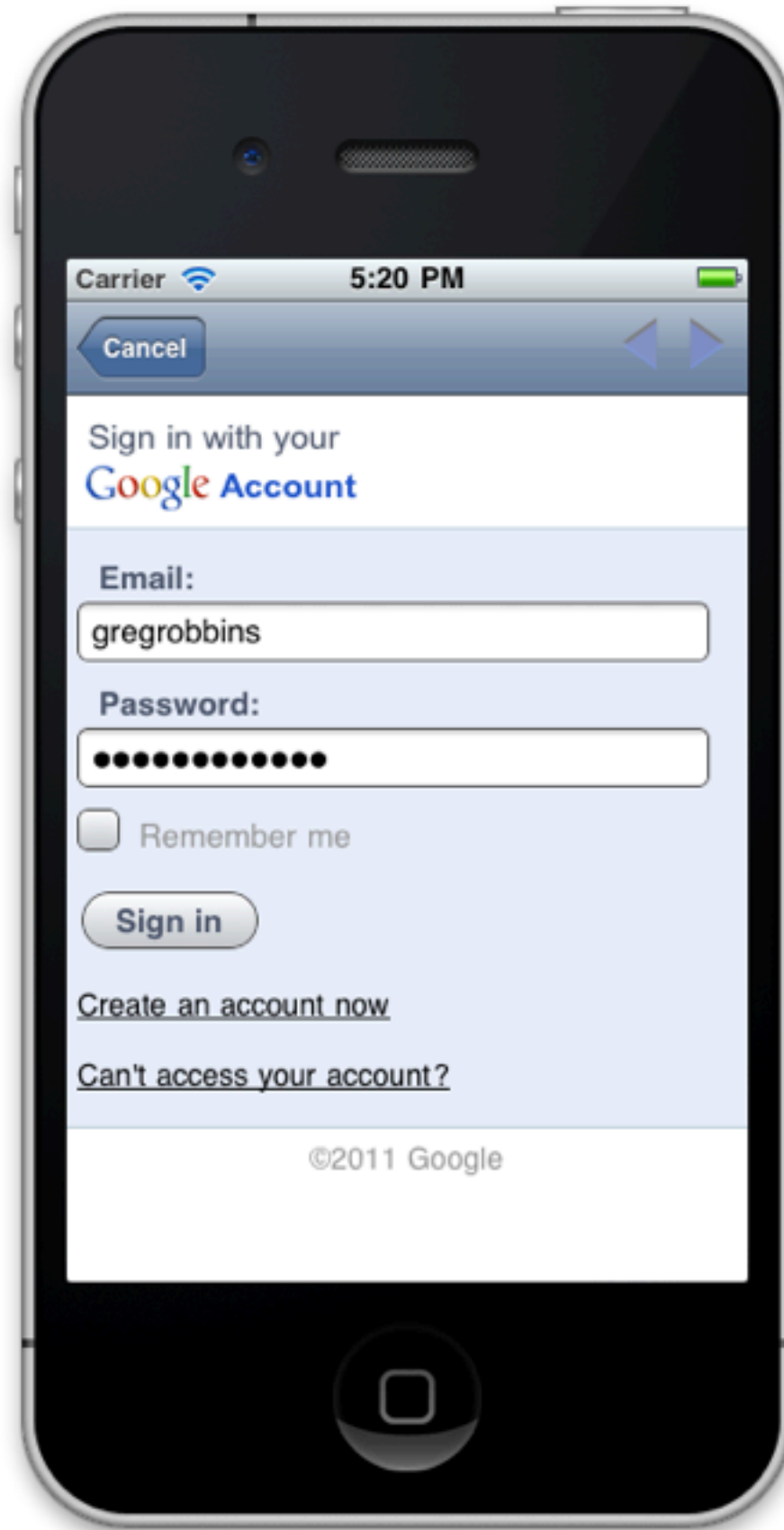
# OAuth for Installed Apps on Windows



(User experience of some Google apps)

Coming soon: open-source release of library

# OAuth on iOS





# Browser vs. Web View

Did you notice the difference?

# When to Use Browser vs. When to Use Web View

## Rule #1: Use an external browser

- User's cookie already available
- Password hidden from app
- User's special auth plugins already installed
- Working password manager for browser
- Server fingerprinting/risk-based mechanisms won't raise a red flag

## Rule #2: If that doesn't work, use a web view

- e.g. iOS suspends your app when you launch the browser, maybe someone else will grab the token

## Rule #3: Sometimes, use a web view anyway

- e.g. when you know the user has never logged into SP on this device before (Android account setup)

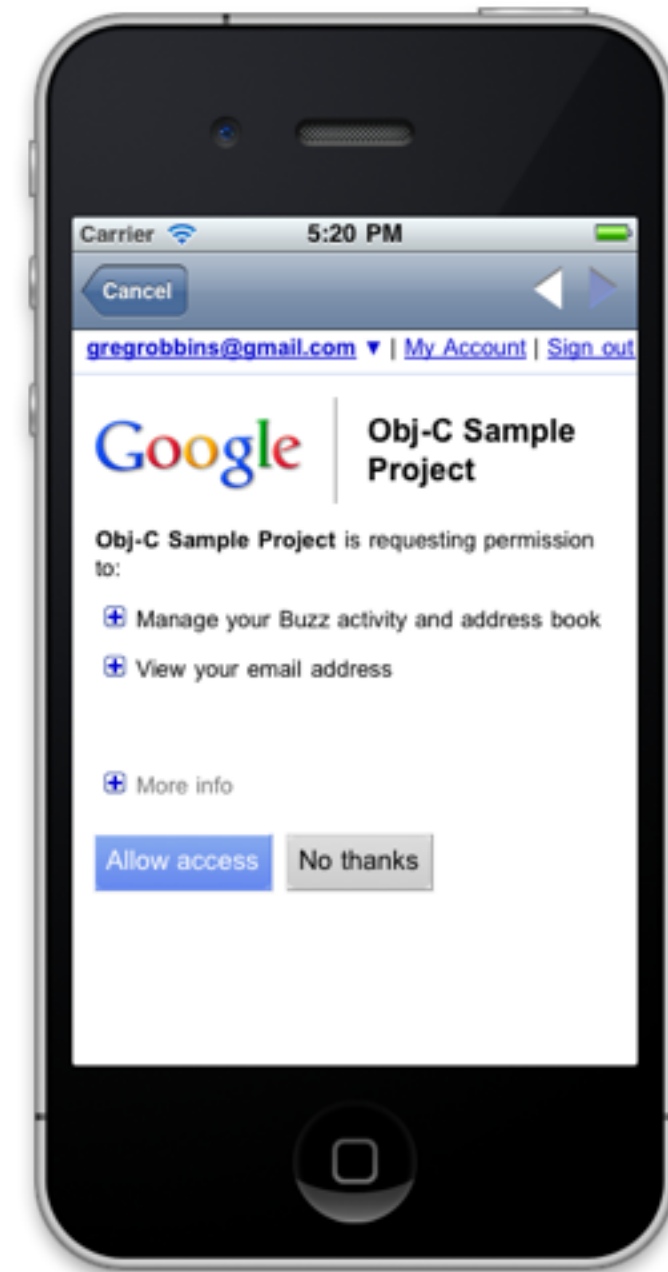
# HOWTO: Windows/OS X



- Recommended: scrape code from window title
  - Set redirect to `"urn:iETF:wg:oauth:2.0:oob"`
  - Redirects to page at Google that puts code/token in window title
  - Native app cycles through windows, looks for code/token
- Firewall issues with `http://localhost:port` (often blocked)
- Reliability problems with `customscheme://`

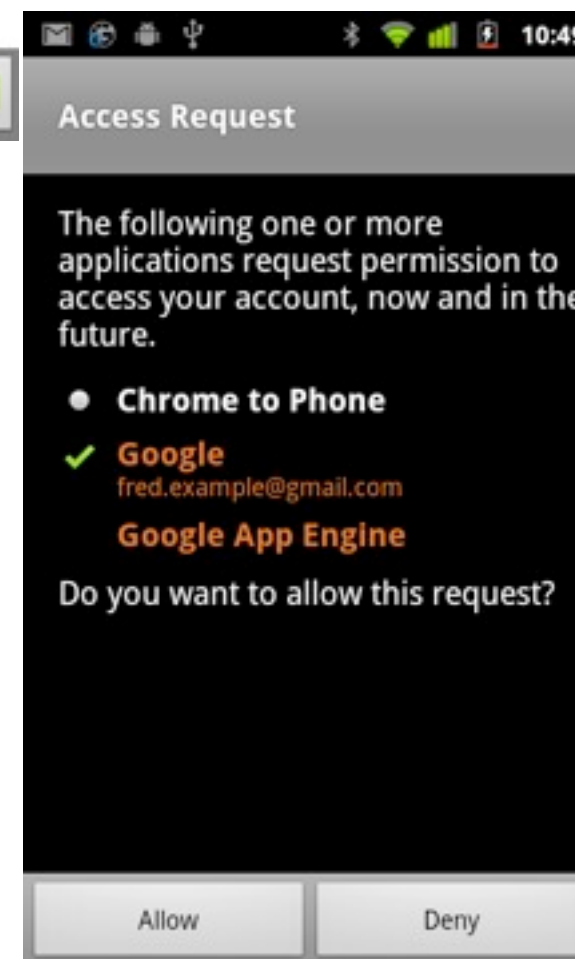
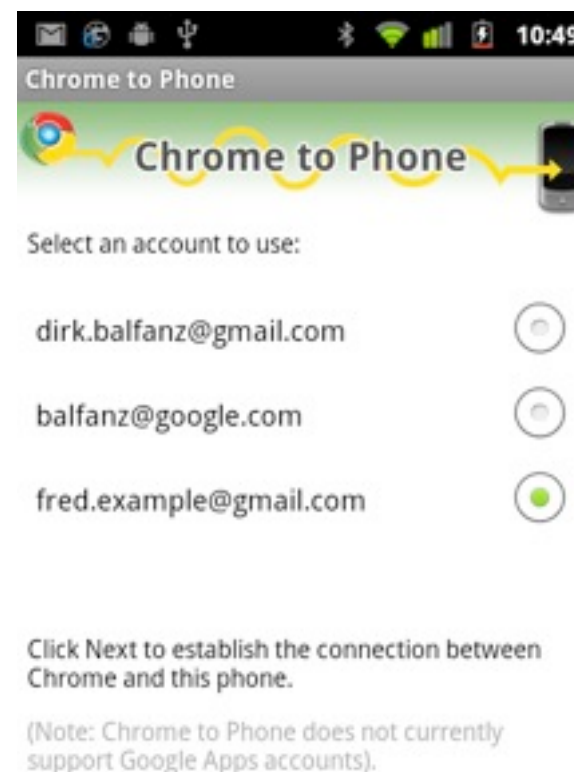
# HOWTO: iOS

- Library available
  - <http://code.google.com/p/gtm-oauth2/>
  - OAuth 1 version also available
- Uses WebView
  - intercepts redirect to `redirect_uri` to obtain token
- Coming Soon:
  - API for requesting tokens
  - Will leverage Google Mobile App (as "account manager") and APIs console



# OAuth on Android

- Phone has user credentials
- Apps never need to ask for user credentials or do OAuth "dance"
- Instead, ask AccountManager for token
- Today: token is "clientlogin" token (scoped to Google service)
- Coming soon:
  - Token can be OAuth 2.0 token (scoped by API scopes)
  - May be required to register app with APIs Console





# HOWTO: OAuth Device Flow

Suitable for awkward or non-existing input methods

- Unlike Android, doesn't have user credentials

1. Device displays "activation code," approval URL
2. User goes to approval URL on PC, enters code
3. Device can pick up OAuth token in return for activation code

Ready for testing—Contact [oauth-device-flow@google.com](mailto:oauth-device-flow@google.com) for details



# OAuth for XMPP & IMAP/POP/SMTP

- OAuth support for IMAP/SMTP now
  - <http://code.google.com/apis/gmail/oauth>
  - Instead of password, put OAuth message/token
  - OAuth 1 today, OAuth 2.0 in next few months
- XMPP
  - OAuth 2.0 support coming soon
- SASL + OAuth 2.0 standard
  - SASL = common auth layer for XMPP, IMAP, POP, SMTP
  - Industry discussions, but still being designed

# Other Issues

- Installed apps can't hold onto client secrets
  - Use the OS to authenticate apps (iOS/Android)
  - How do we identify them on approval pages?
    - self-asserted name from APIs Console
- Scopes
  - URLs that represent permissions/resources
    - <https://www.googleapis.com/auth/contacts>
  - Google has many (~100)

# Summary

- ClientLogin breaks for many different use cases
  - OpenID/SAML/2factor/suspended...
- Application-Specific Passwords are a stop-gap
  - work for some, but not all, of the above use cases
- OAuth 2.0 is the Official Solution
  - different libraries for different platforms

# Learn More

- Related I/O session

- Identity and Data Access (OpenID & OAuth)
- Day 2, 1:45pm, Ryan Boyd

- Online references

- Search for "OAuth 2.0 Google"
- <http://code.google.com/apis/accounts/docs/OAuth2.html#IA>

# ClientLogin #FAIL

## Q&A



**feedback** <http://goo.gl/2Qxx6>

**hashtags** #io2011 #GoogleAPIs