Feedback: **http://goo.gl/DpUBh**
#io2011 #TechTalk

Identity and Data Access: OpenID & OAuth

Ryan Boyd @ryguyrg
https://profiles.google.com/ryanboyd
May 11th 2011

Feedback: **http://goo.gl/DpUBh**
#io2011 #TechTalk

Agenda

1. Terminology
2. OpenID and the Google Identity Toolkit
3. Mobile Authentication
4. OAuth for Individuals
5. OAuth for Businesses
6. The Future!
7. Resources and Q&A

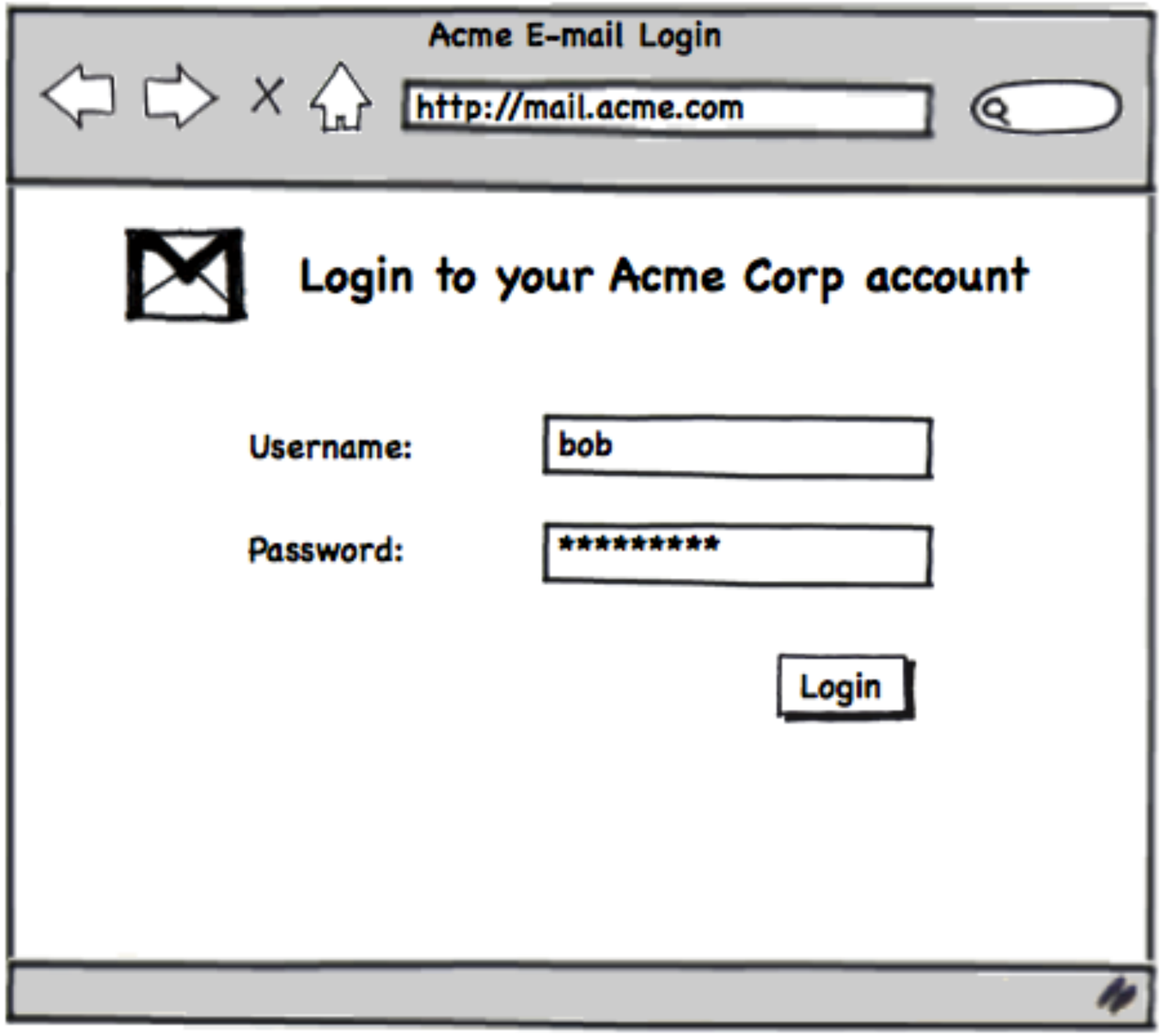
Terminology

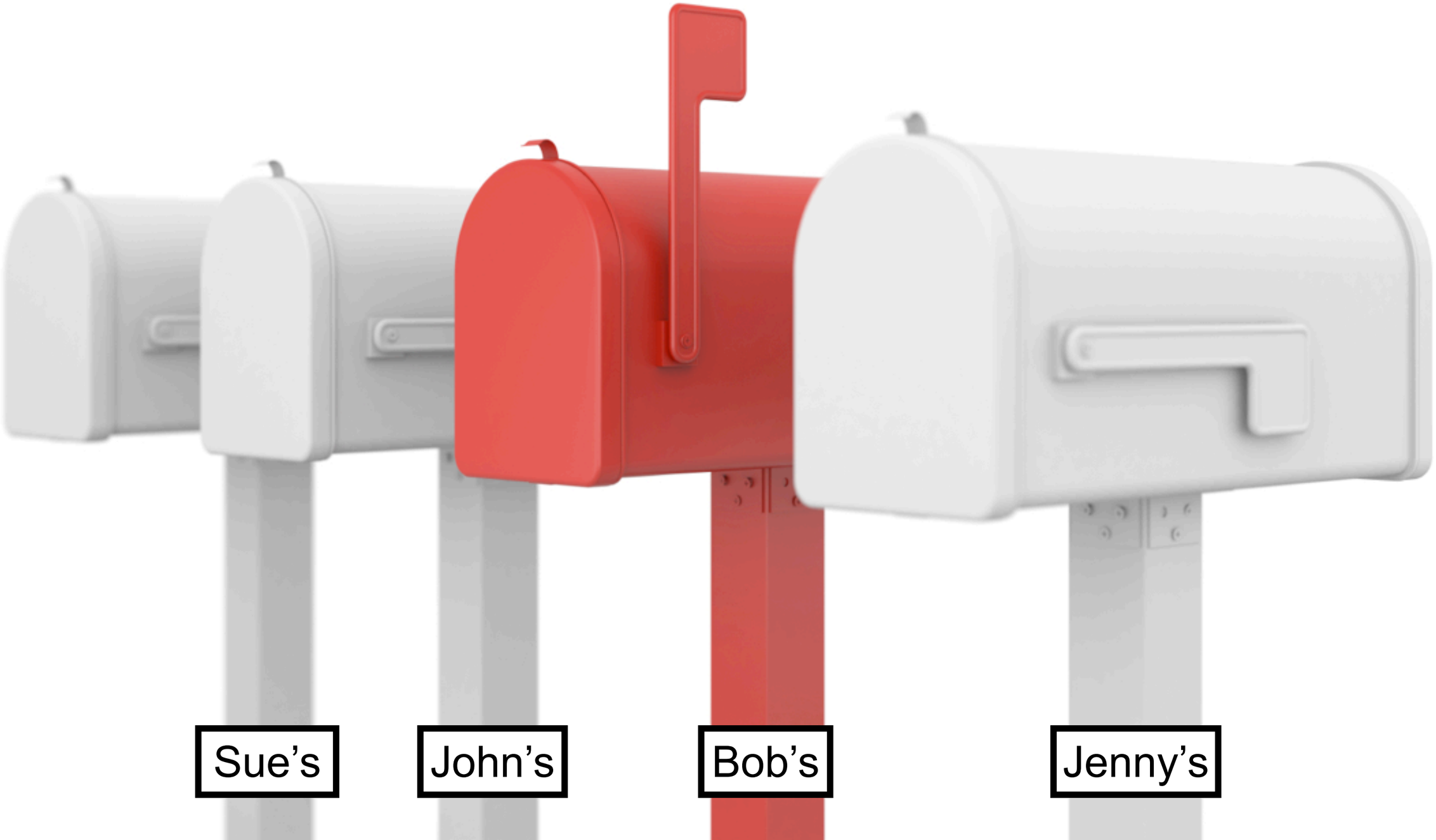
Authentication

Authorization

Bob

Administrative Assistant
@ Acme Corporation


Acme E-mail Login
http://mail.acme.com
Login to your Acme Corp account
Username:
bob
Password:
****
Login


Sue’s
John’s
Bob’s
Jenny’s

Bob is
Authenticated
to his Acme Corp
account and has
Authorized
access to his
mailbox

Authentication and Authorization in Context

1. **OpenID** for **Authenticating** a user visiting a web site

2. **OAuth** for **getting Authorized access** to a user's data stored elsewhere

Authenticating users via OpenID

OpenID: Terminology

Identity Providers (IdP)

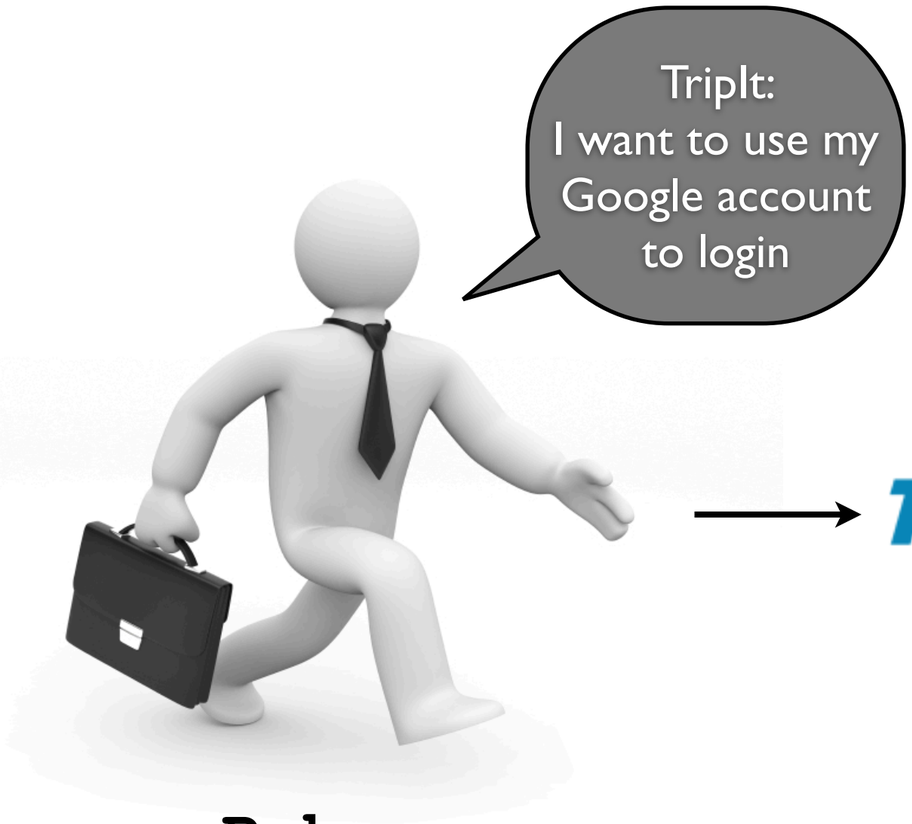
Relying Parties (RP)

How does Federated Identity Work?

Bob

TripIt

How does Federated Identity Work?

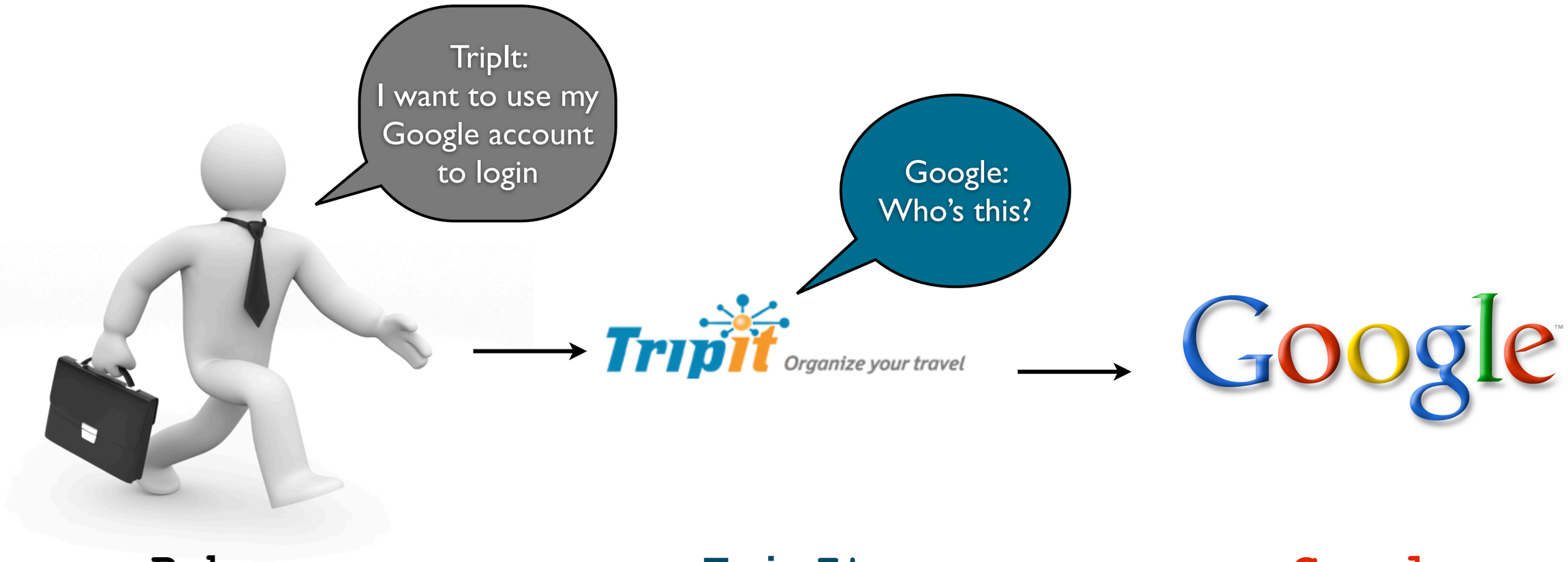


Bob

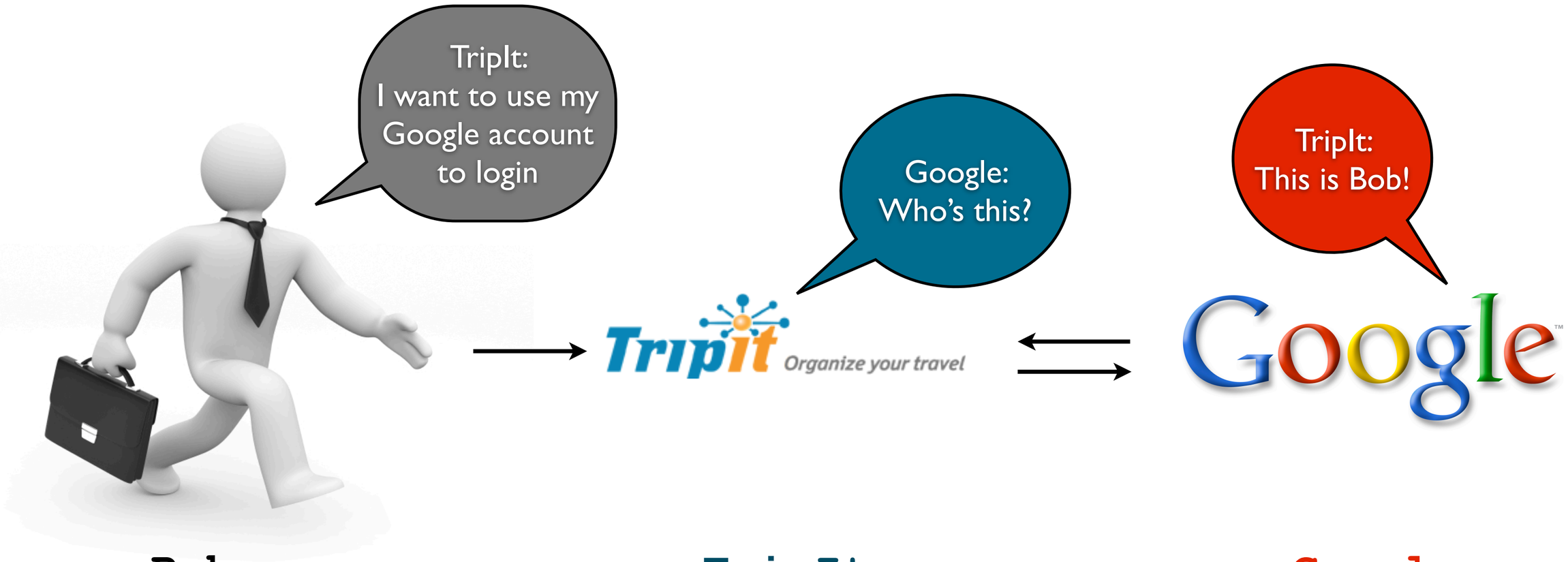
TripIt

Google

How does Federated Identity Work?



How does Federated Identity Work?



Bob

End-User

TripIt

Relying Party

Google

Identity Provider

S U C C E S S !!

Bob is
Authenticated
to his
TripIt account
using his
Google identity

OpenID: Why?

Users can login to all sites

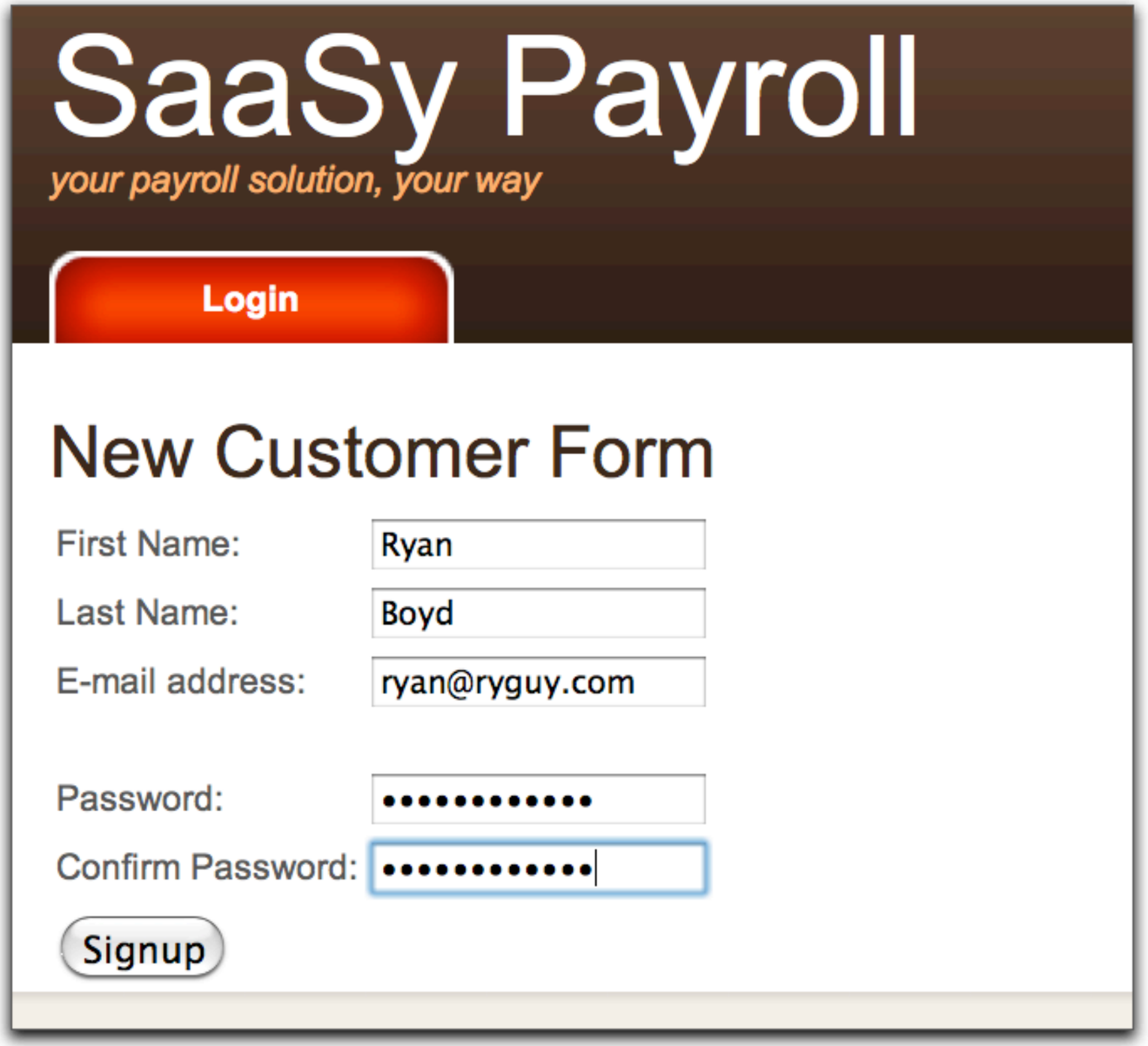
using their existing accounts

"OpenID is a **safe, faster**, and **easier** way to log in to web sites."

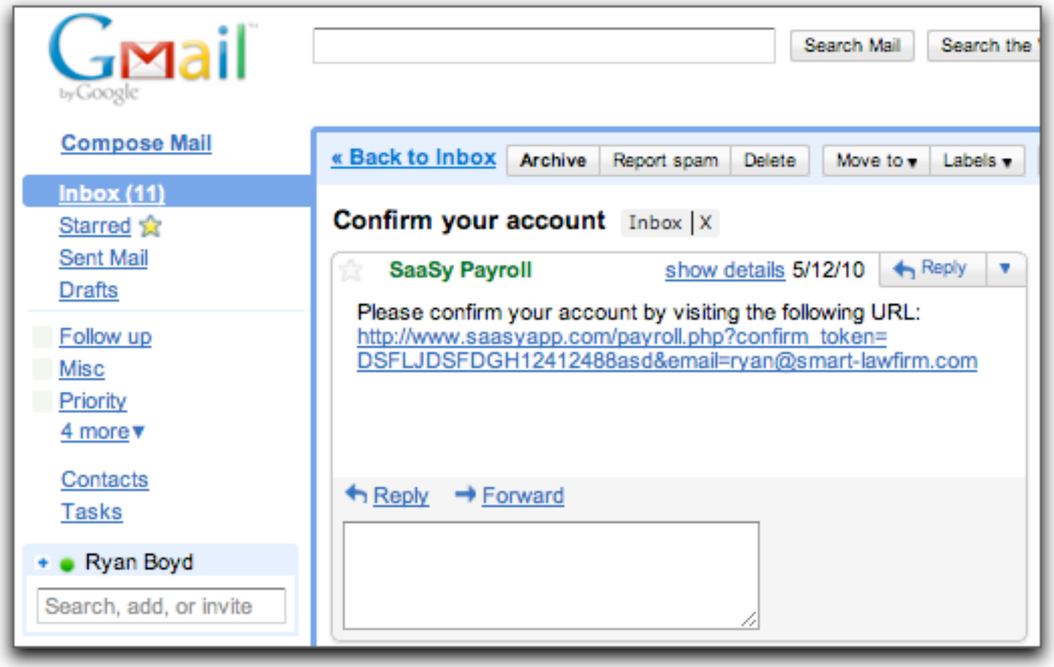
openid.net

Faster and Easier

Traditional Signup Form



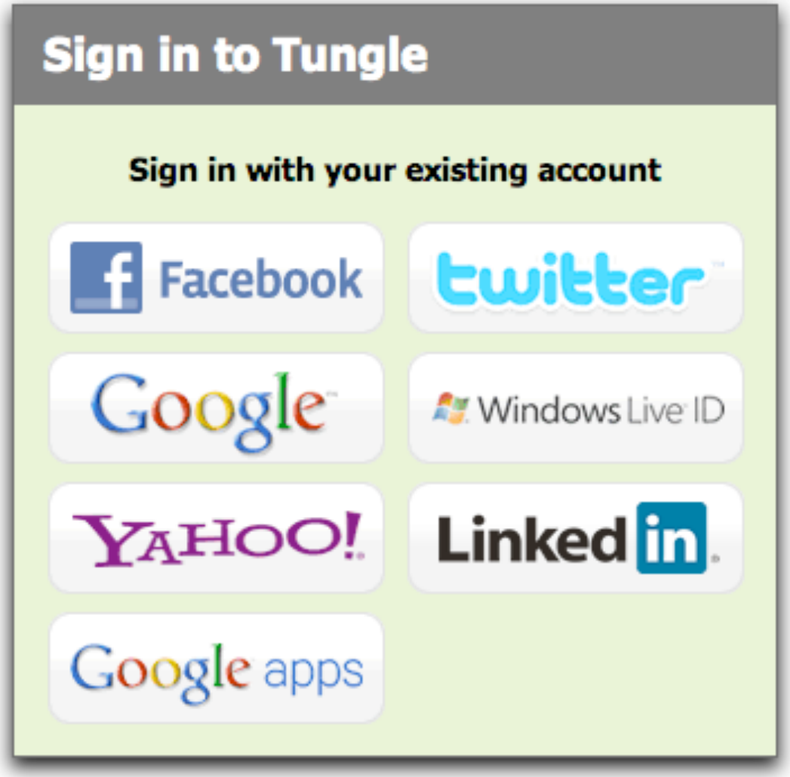
Traditional Signup Form - e-mail confirmation



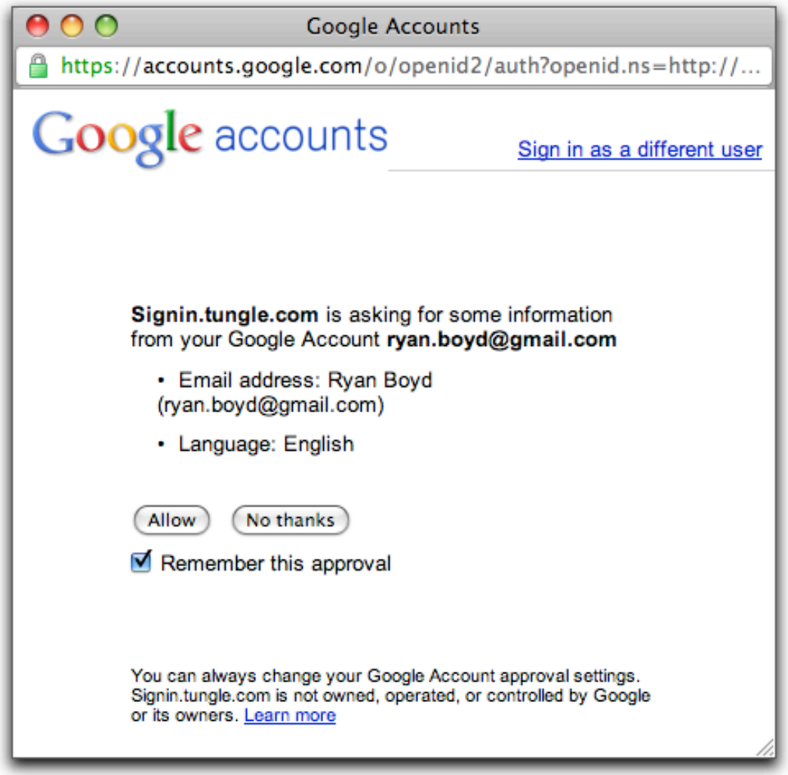
50 keystrokes

3 mouse clicks

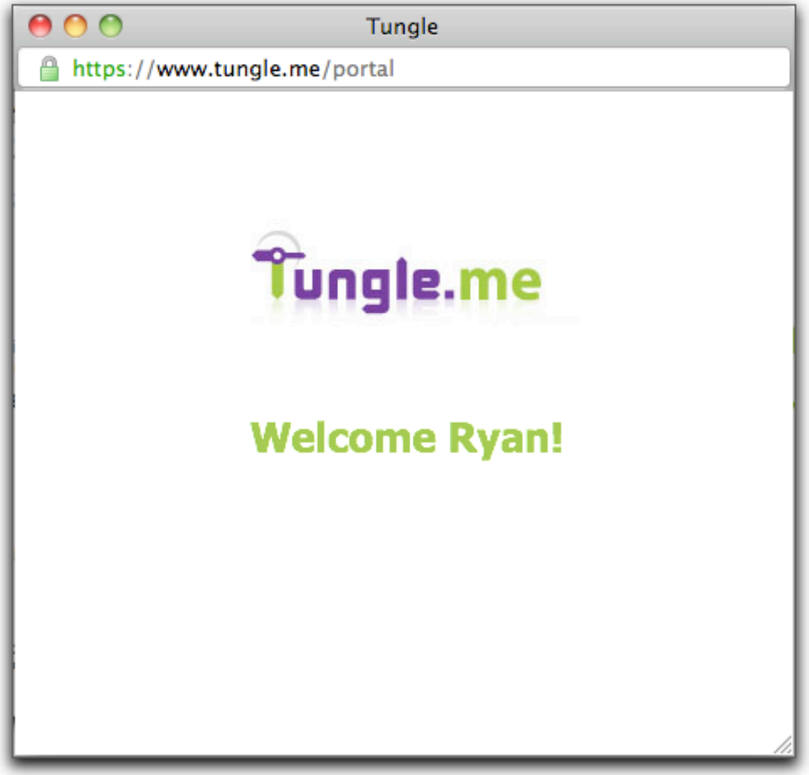
Improved UX with OpenID



1



2



3

0 keystrokes
2 mouse clicks

Safer

Ways OpenID is Safer

1. One username and password
2. Password can be ultra-secure
3. Password is only provided to Identity Provider
4. Two-factor auth and other protections

Safer, Faster and Easier

The user only provides their ultra-secure username/password to their identity provider

The user is often already logged into their identity provider

The user doesn’t need to create, maintain and enter a password on every site

OpenID: becoming a Relying Party

OpenID is easy to implement

But not easy ENOUGH

Why? Edge Cases!

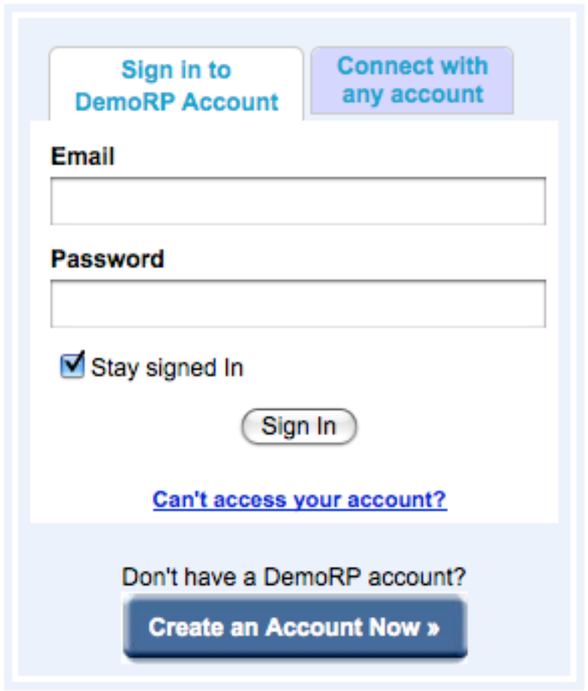
- Existing user wants to switch to using OpenID
- OpenID user wants to switch back to a password
- User changes their e-mail address
  - New address matches another account
- Handling deleted/suspended accounts

See http://www.openidsamplestore.com/

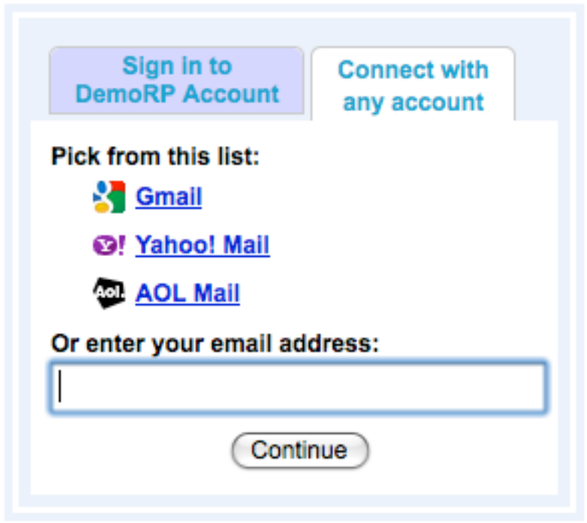
Introducing ...

Google Identity Toolkit

Google Identity Toolkit



Google Identity Toolkit



Provides:

- JavaScript UI Widgets
- Client Libraries
- Code on Google's servers

Supports:

- Signup and/or Login
- Multiple Identity Providers:
  - Gmail (including Google Apps)
  - AOL Mail
  - Yahoo Mail
  - Hotmail

Demo Relying Party site usin

https://easyrpsample.appspot.com/login.jsp

Demo RP Site using **GIT**

Productivity for developers, performance for users

GIT provides developers with a powerful set of API endpoints to do federated login, attribute exchanges. This allows desktop, mobile, and web application developers to integrate with different identity provider, and facilitate the user to sign-up and sign-in on different sites.

Try the Demo

This site supports federated login by leveraging the GIT.
Try federated login to this site by the 2-tab login widget on the right.
You can also try the Create Account Wizard to import attributes from IDP.

Get Started

Walk through the first steps needed to integrate GIT into your application. From there, work through the fundamentals of GIT development with an in-depth tutorial.

Read the Docs

Everything you need to know about how to use GIT.

Sign in to DemoSite account | Connect with any account

Email

Password

Stay signed In

Sign In

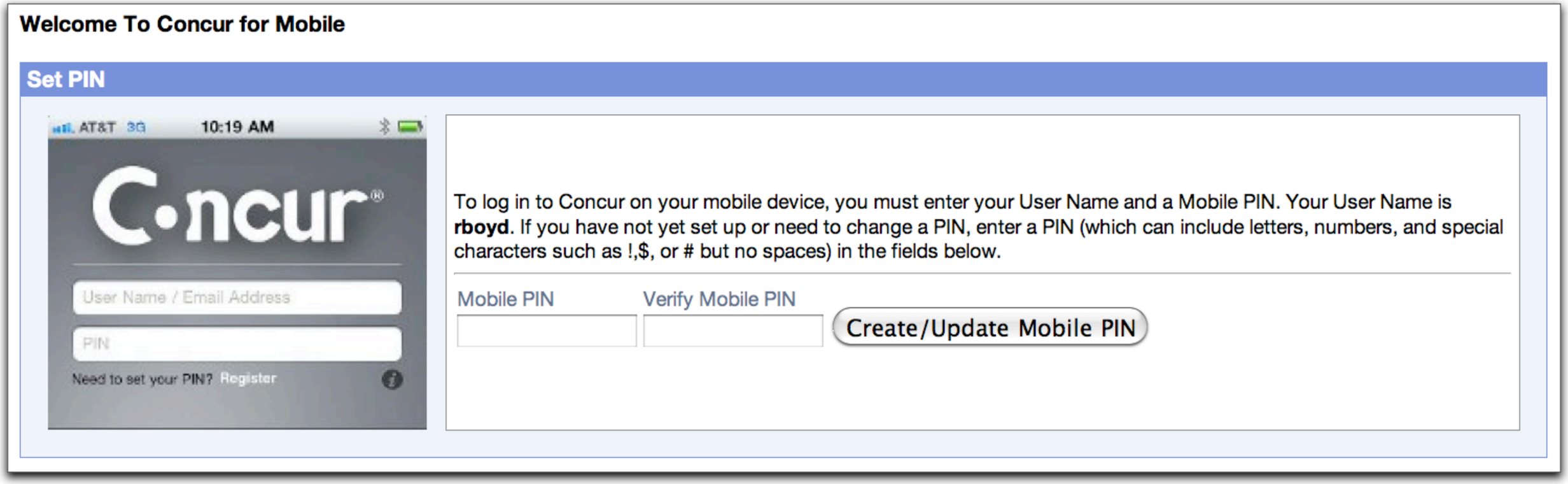
Can't access your account?

Don't have a DemoSite account?

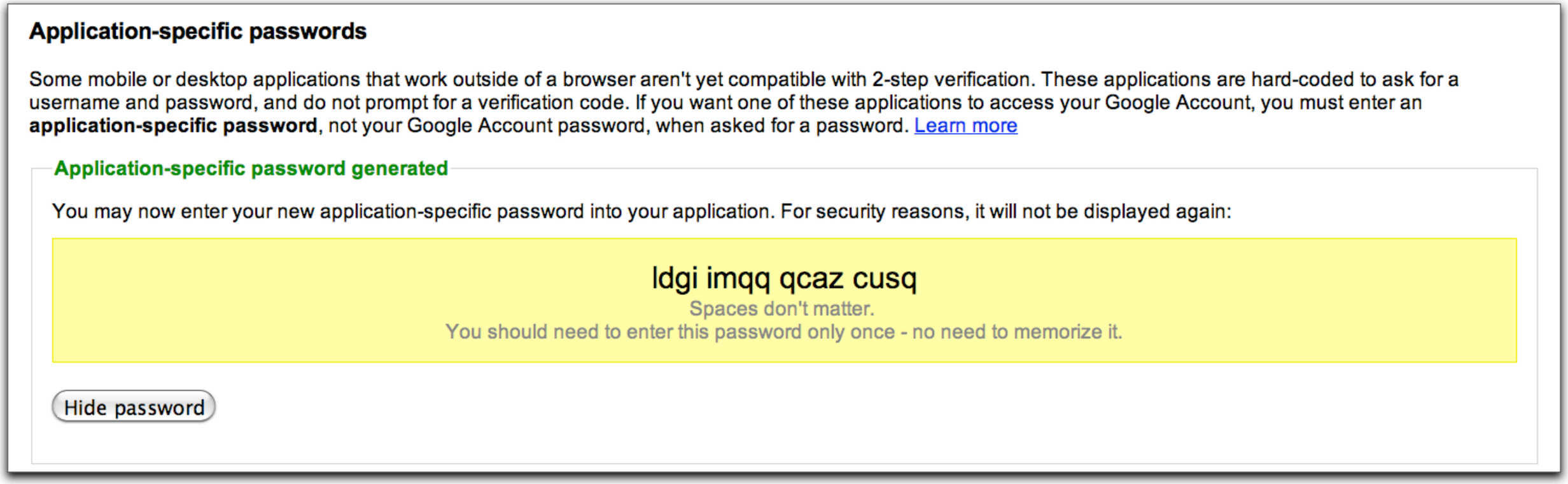
Create an Account Now »

Authenticating Users on Mobile Devices

Allow users to create a ‘Mobile Password’



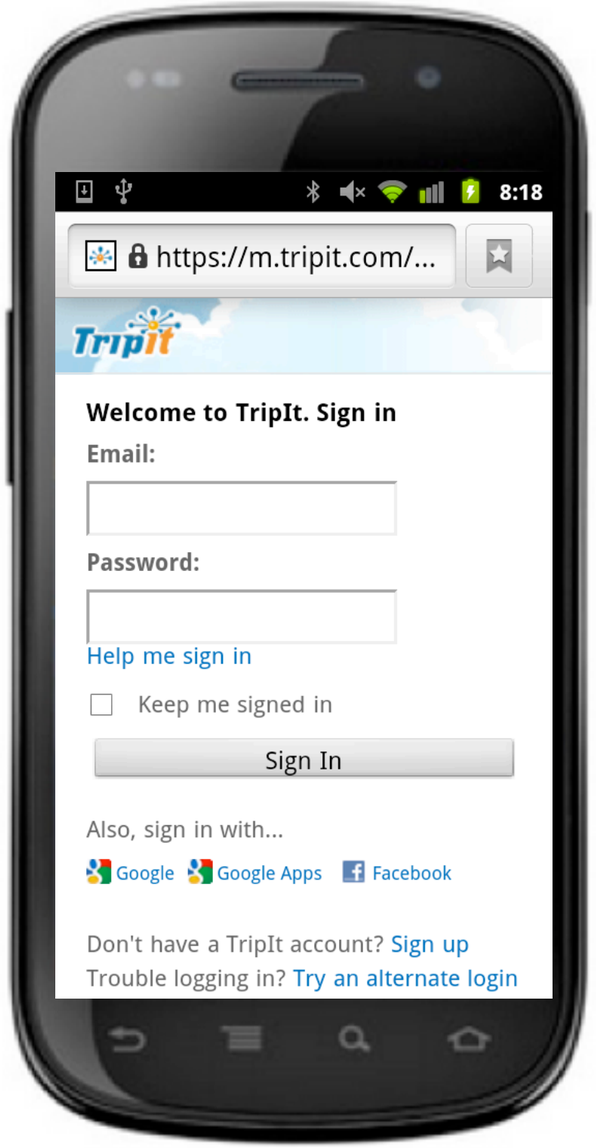
Generate a ‘Mobile Password’



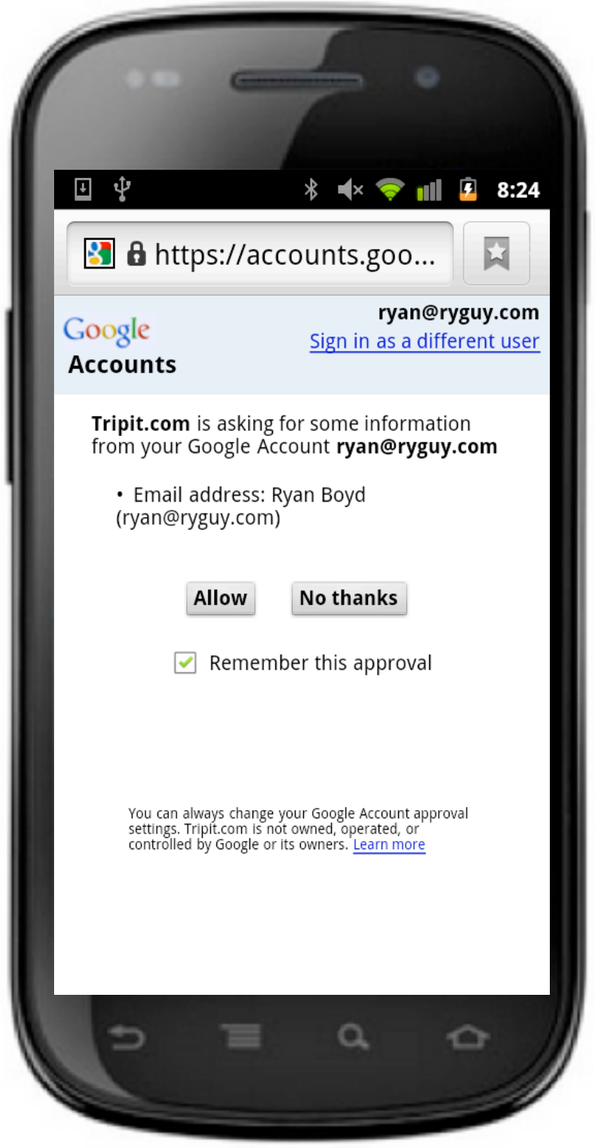
OpenID!


8:15
TripIt
Organize your travel
Sign in to TripIt
More about TripIt

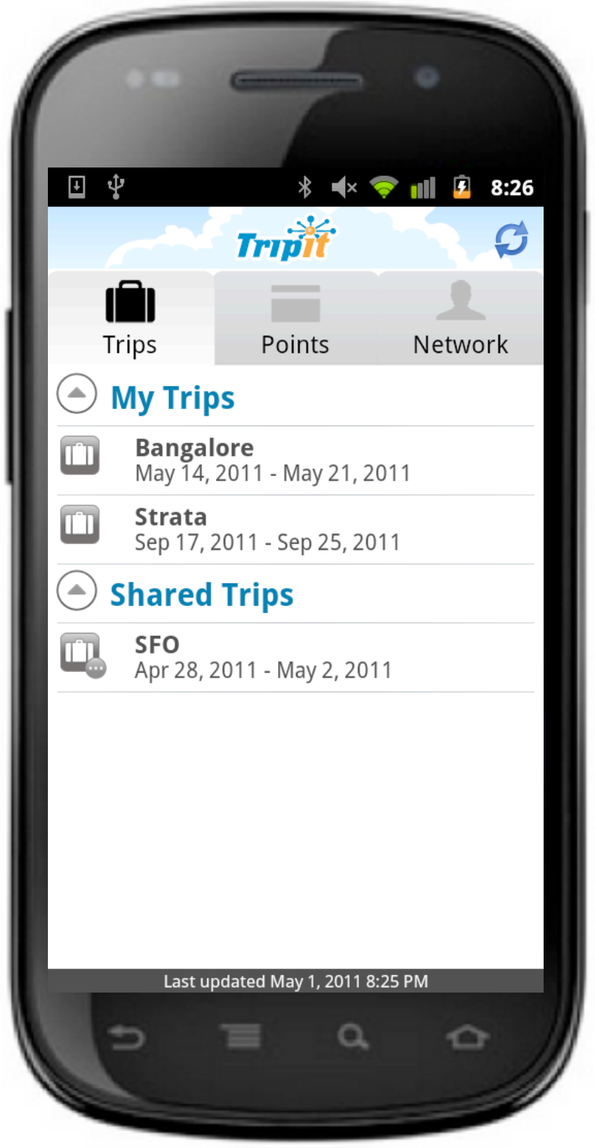
1


8:18
https://m.tripit.com/...
TripIt
Welcome to TripIt. Sign in
Email:
Password:
Help me sign in
Keep me signed in
Sign In
Also, sign in with...
Google
Google Apps
Facebook
Don't have a TripIt account? Sign up
Trouble logging in? Try an alternate login

2


8:24
https://accounts.goo...
Google
Accounts
ryan@ryguy.com
Sign in as a different user
Tripit.com is asking for some information from your Google Account ryan@ryguy.com
• Email address: Ryan Boyd (ryan@ryguy.com)
Allow
No thanks
Remember this approval
You can always change your Google Account approval settings. Tripit.com is not owned, operated, or controlled by Google or its owners. Learn more

3


8:26
TripIt
Trips
Points
Network
My Trips
Bangalore
May 14, 2011 - May 21, 2011
Strata
Sep 17, 2011 - Sep 25, 2011
Shared Trips
SFO
Apr 28, 2011 - May 2, 2011
Last updated May 1, 2011 8:25 PM

4

Getting authorized data access via OAuth

OAuth: Why?

35+ APIs

What data can your app access?

Contacts

Calendar

Picasa Web Albums

YouTube

Why not just ask for the user's password?

OAuth: Terminology & Concepts

OAuth Terminology & Concepts

Protected Resource

- Resides on server
- Requires authorization

Resource Owner

- Owns protected resource
- Approves access

OAuth Terminology & Concepts

Server

- Holds the protected resource

SaaSy Payroll
your payroll solution, your way

Client

- Web application
- Needs access to the protected resource

Who Owns the Data?

Individual owns the resource

- Individual user owns their own data, and decides whether to grant access

Company owns the resource

- Data is owned by a company and access is granted by IT guardians

OAuth Individual Grant Use Case (via OAuth 2.0)

Who Owns the Data?

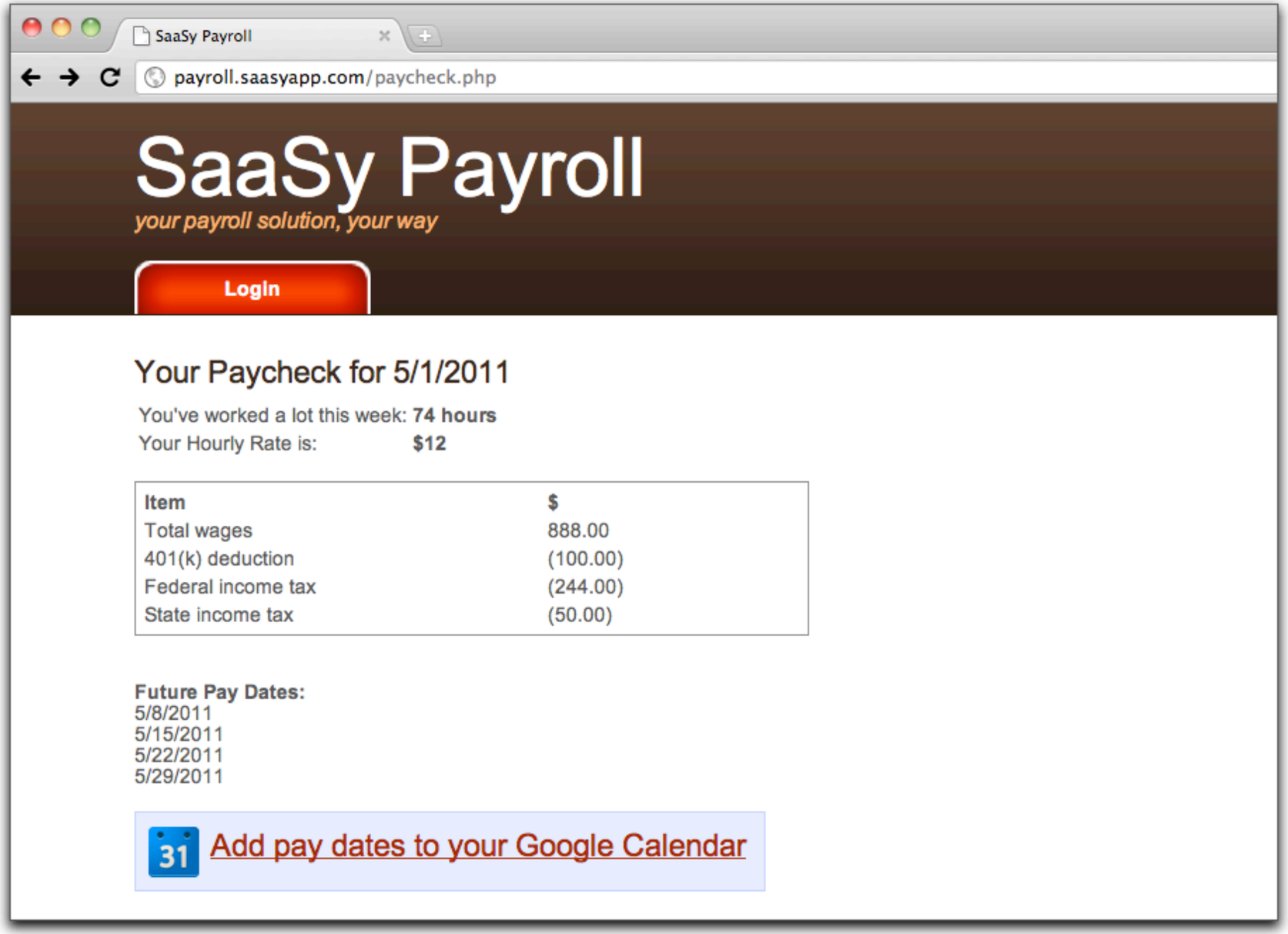
Individual owns the resource

- Individual user owns their own data, and decides whether to grant access

SaaSy Payroll

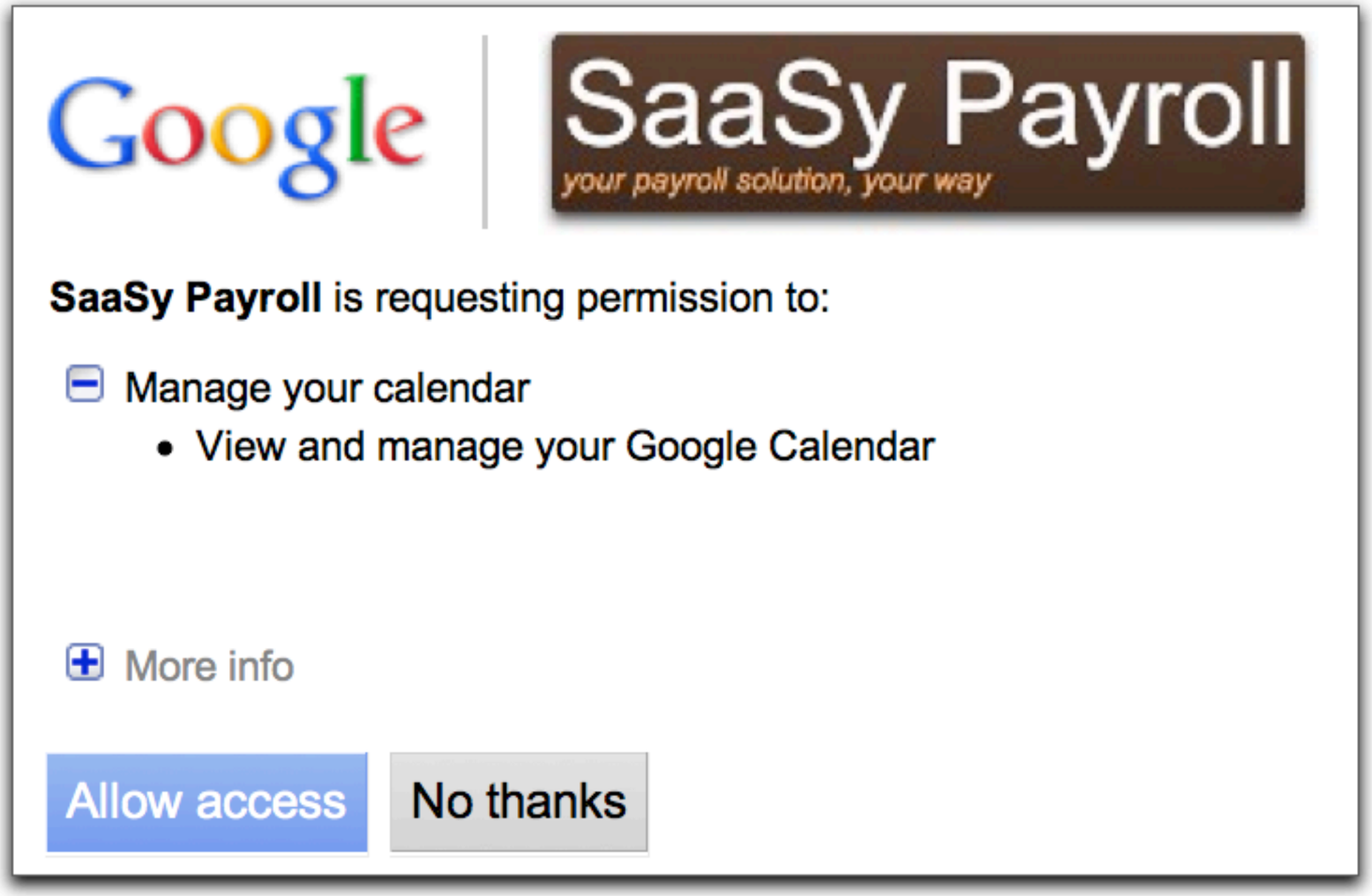
SaaSy Payroll

1



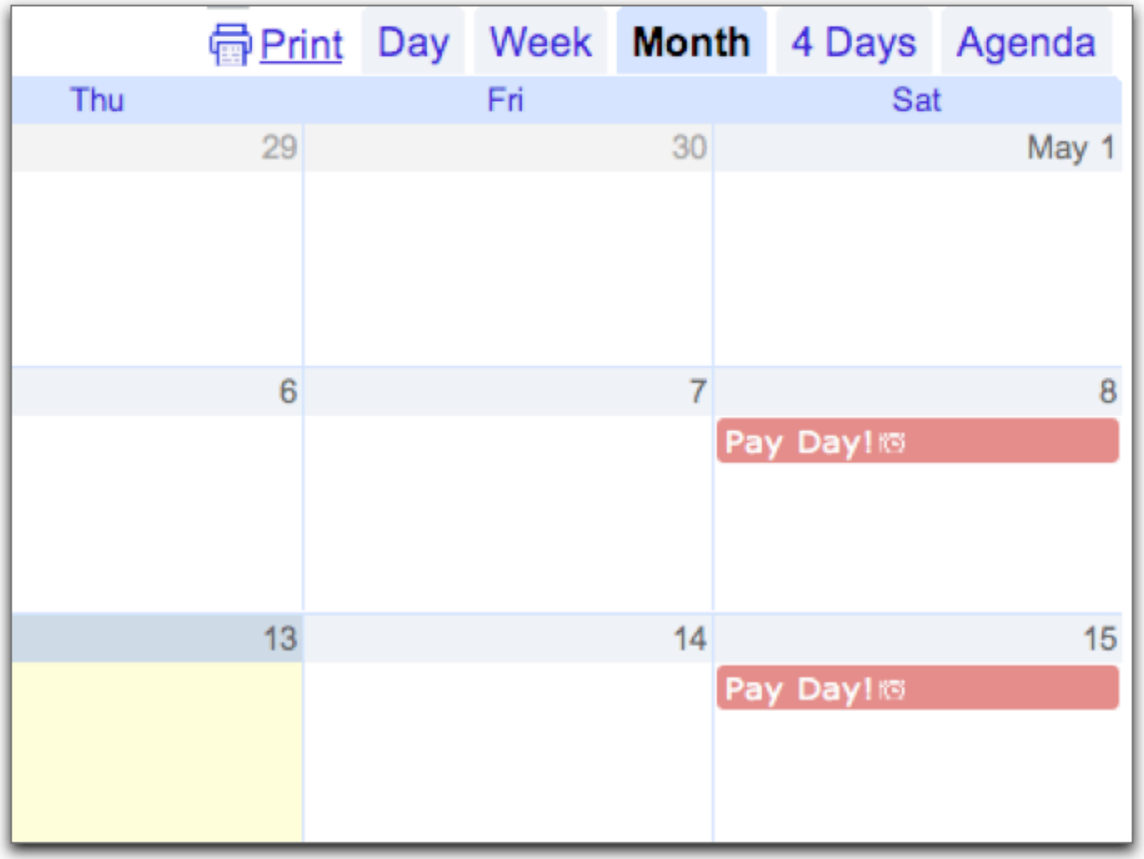
Access Control Grant

2



Payroll on the Calendar

3



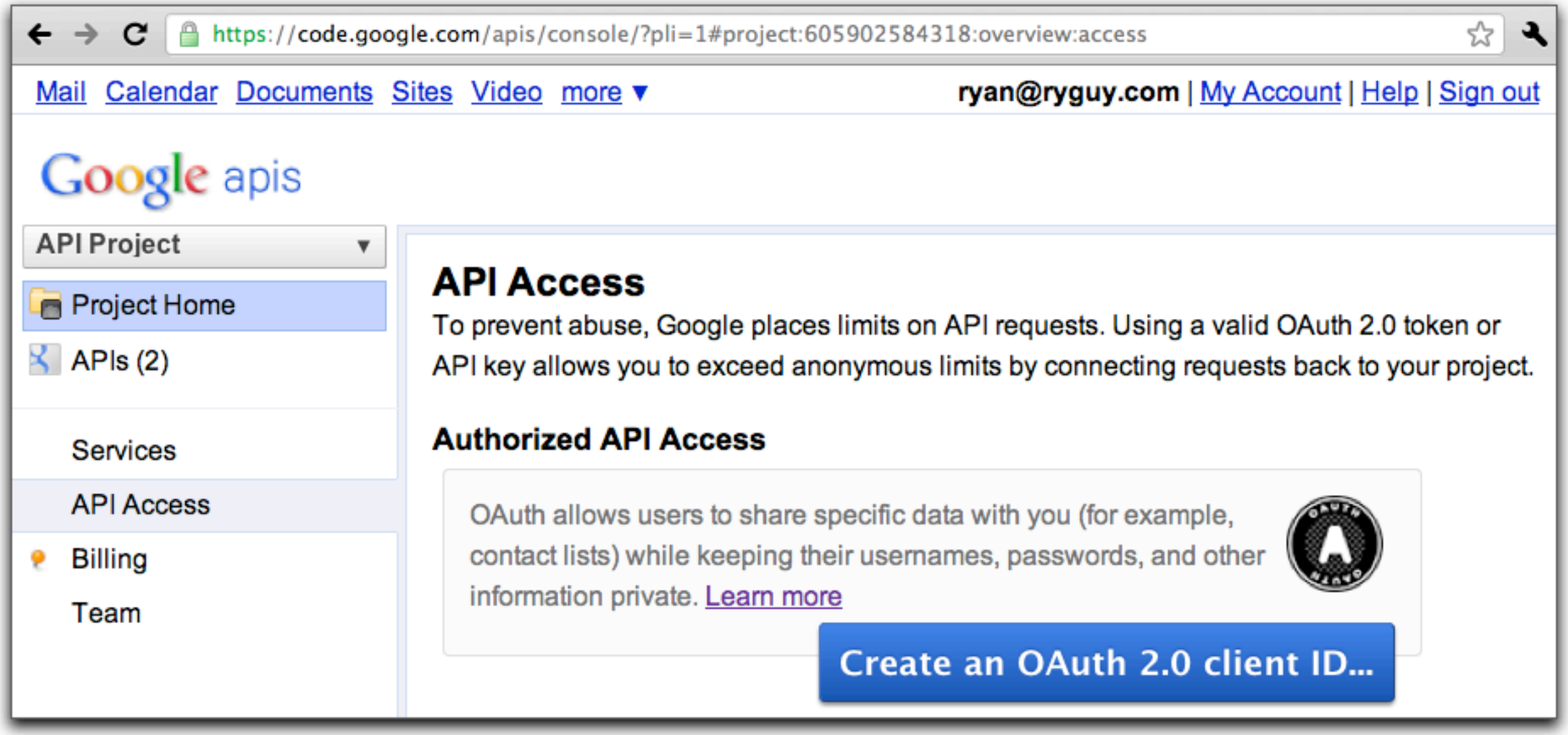
Ryan's Calendar

OAuth 2.0 Flow

Registering an Application

0

Developer registers application with Google, gets a **`client_id`** and **`client_secret`**



Granting Data Access

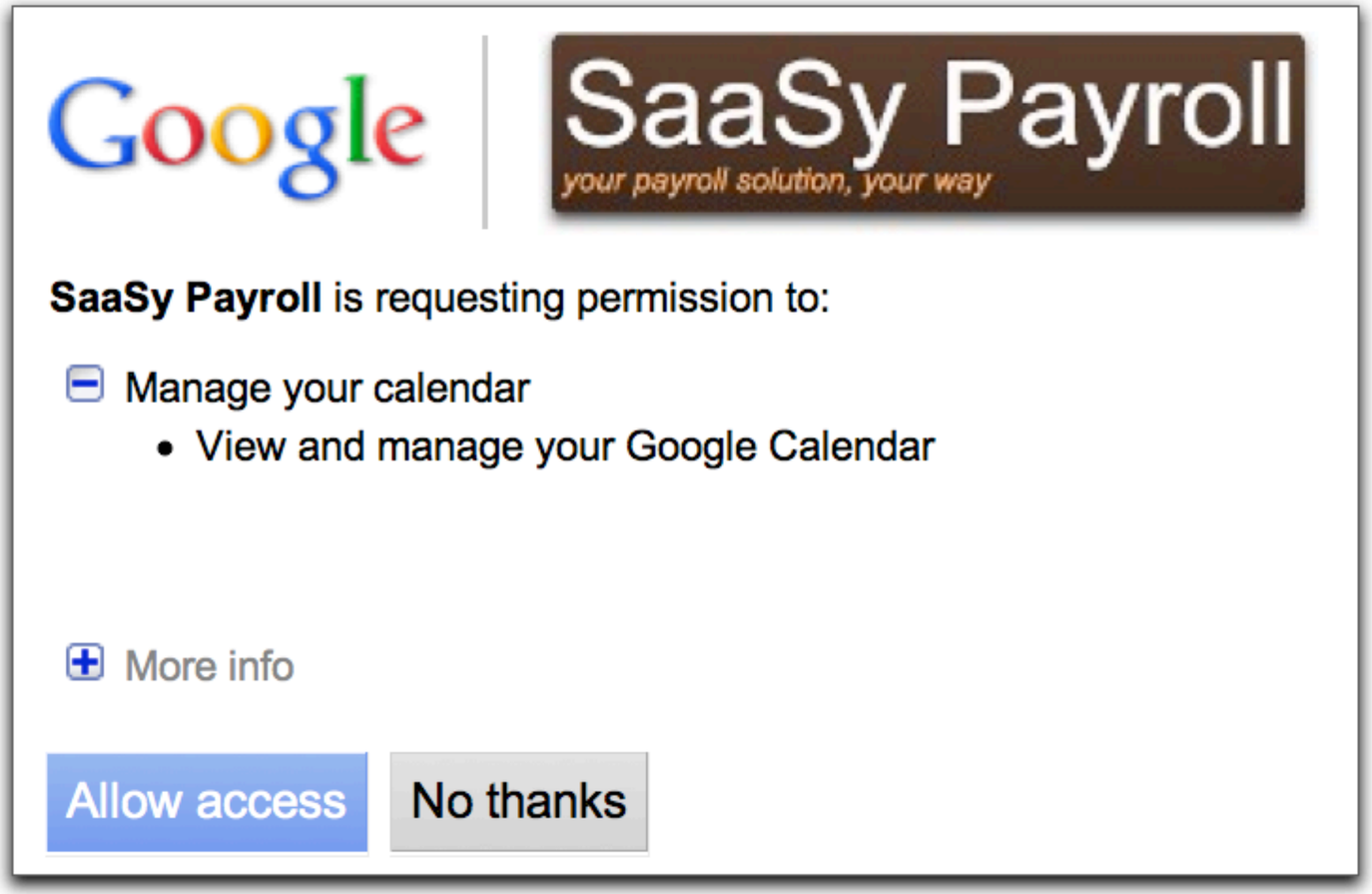
1

Application redirects user to Google, specifying:

`client_id` obtained during registration

`redirect_uri` for user to return to

`scope` or APIs the app needs access to



Obtaining an Access Token and Refresh Token

2

Google redirects the user back to the application's **`redirect_uri`** and includes an **`authorization_code`** in the URL.

```
http://www.saasyapp.com/payroll/back?code=<authorization_code>
```

3

Application performs a HTTP **`POST`** request to Google, including the **`client_id`**, **`client_secret`** and **`code`**. Google returns an **`access_token`** and a **`refresh_token`**.

```
{
"access_token":"1/fFAGRNJru1FTz70BzhT3Zg",
"expires_in":3920,
"refresh_token":"1/6BMfW9j53gdGX-tqf8JXQ"
}
```

Calling an OAuth Protected API

4

Application makes a HTTP **`GET`** or HTTP **`POST`** request to the server containing the protected resource, including the **`access_token`** as a query param or header.

Query-param:

```
https://www.google.com/calendar/feeds/default/private/full?oauth_token=<access_token>
```

Header:

```
Authorization: OAuth <access_token>
```

Refreshing the Access Token

Application performs a HTTP **`POST`** request to Google, including the **`client_id`**, **`client_secret`** and **`refresh_token`**. Google returns an **`access_token`**. Refresh token remains the same, indefinitely until revoked.

```
{
  "access_token":"1/fFAGRNJru1FTz70BzhT3Zg",
  "expires_in":3920
}
```

OAuth 2.0: The Whole Flow

0 — Developer registers application → Google

1 — User visits SaaSy Payroll → SaaSy Payroll

SaaSy Payroll asks user to authorize data at Google

2 — User grants data access to app → Google

Google tells user to return to SaaSy Payroll with **`code`**

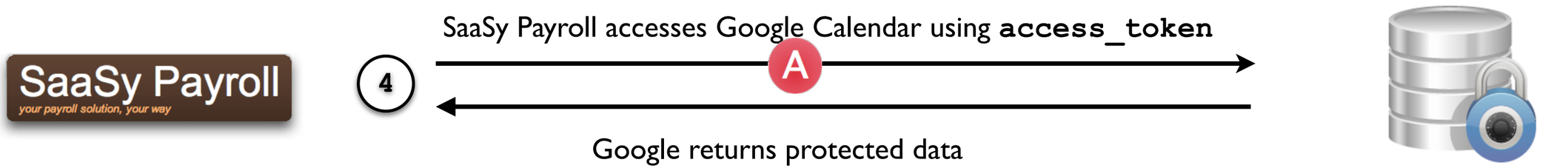
SaaSy Payroll

3 — SaaSy Payroll asks Google for an **`access_token`** → Google

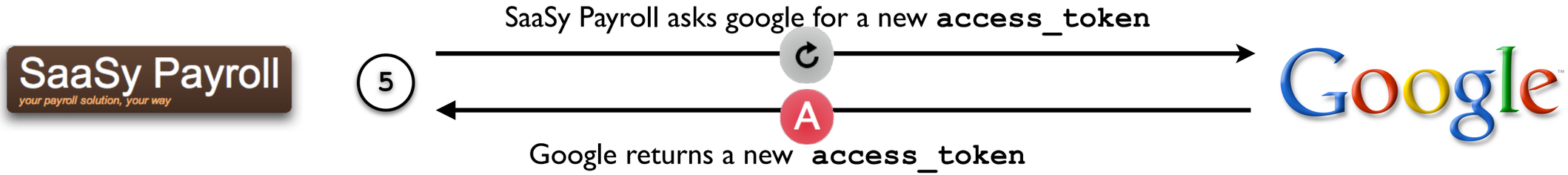
A

Google returns an **`access_token`** and a **`refresh_token`**

The Whole Flow (Continued)



Some time later



OAuth Business Use Case (via 2-legged OAuth 1)

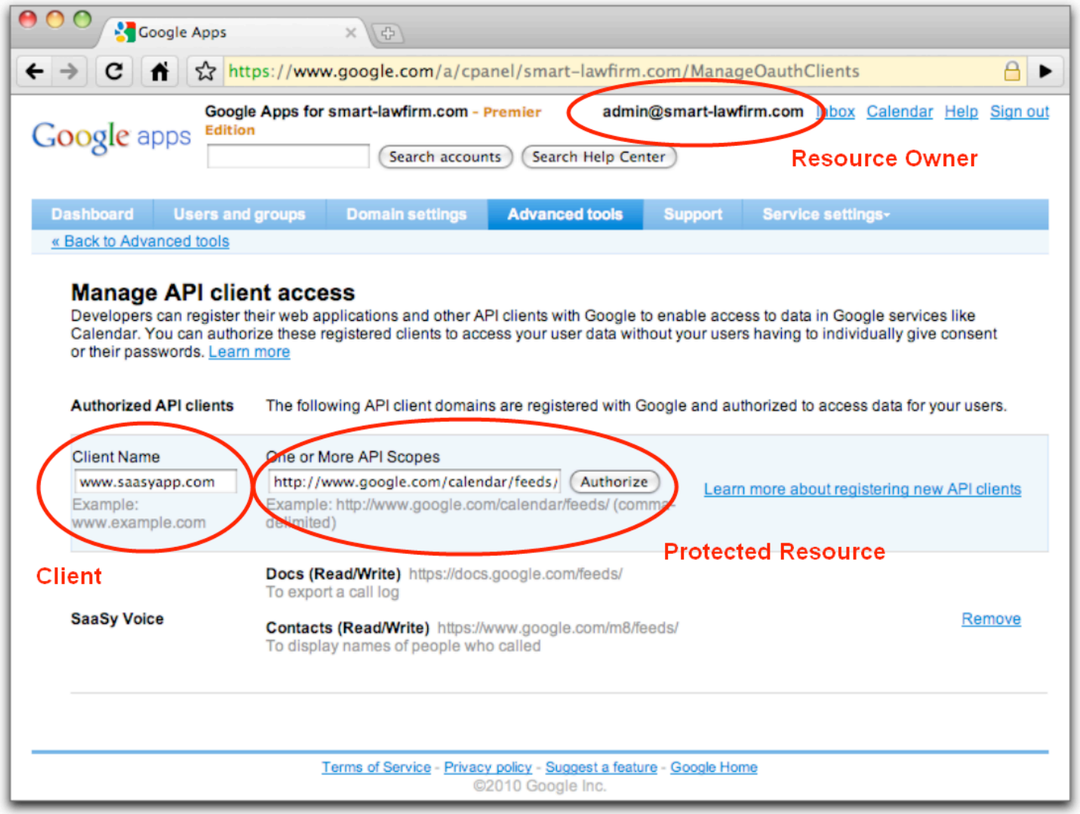
Who Owns the Data?

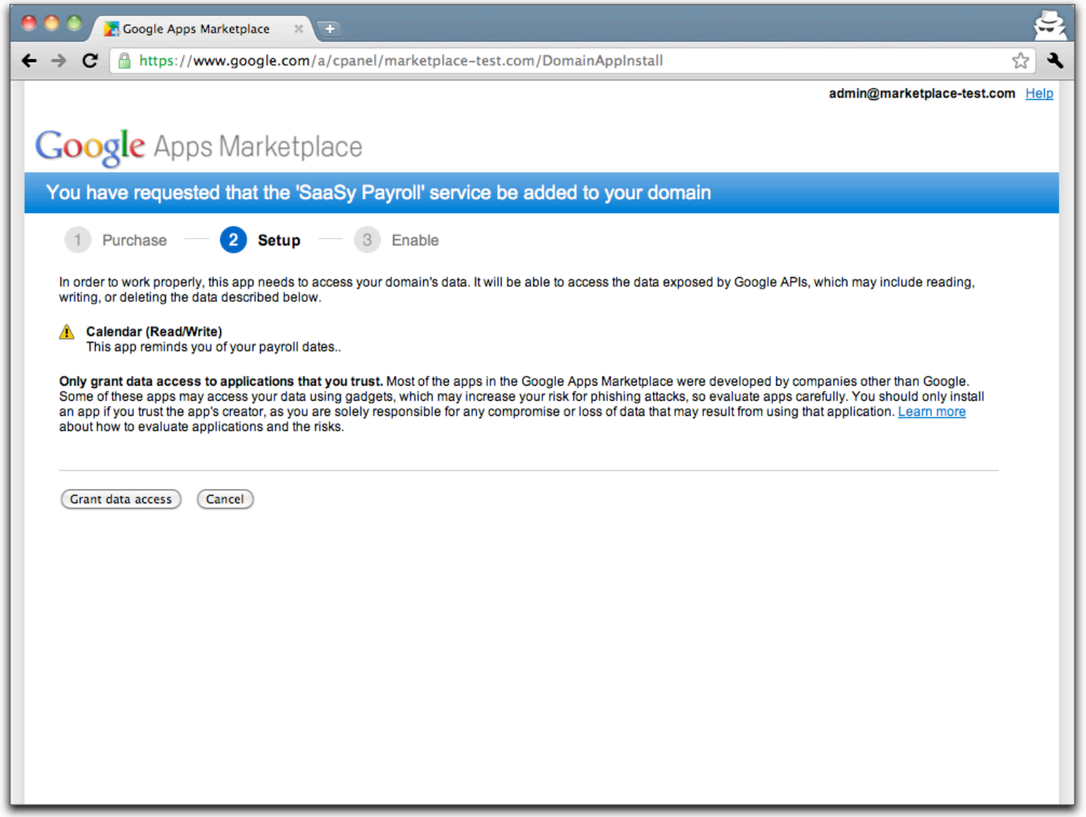
Company owns the resource

- Data is owned by a company and access is granted by IT guardians

Access Control Grant

1



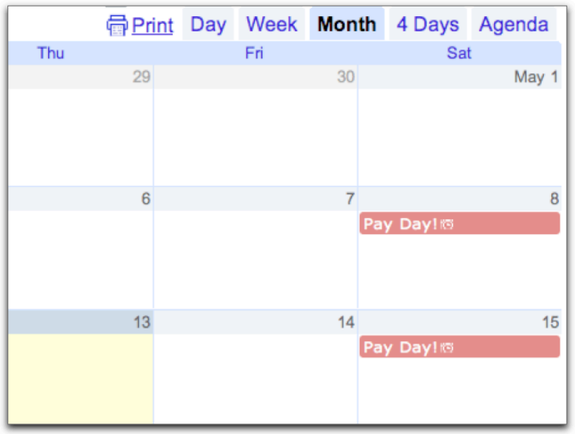


Google Apps Control Panel

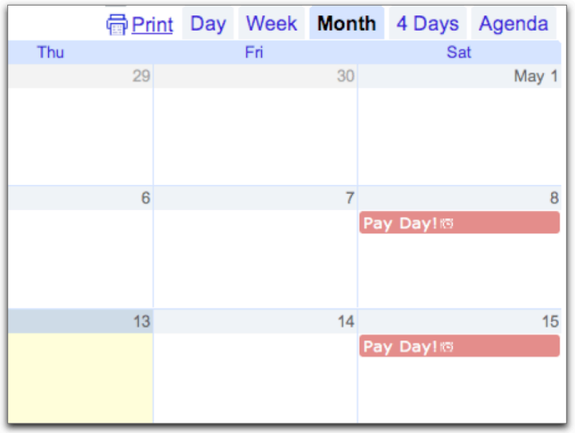
Google Apps Marketplace

Payroll on All Employees' Calendars

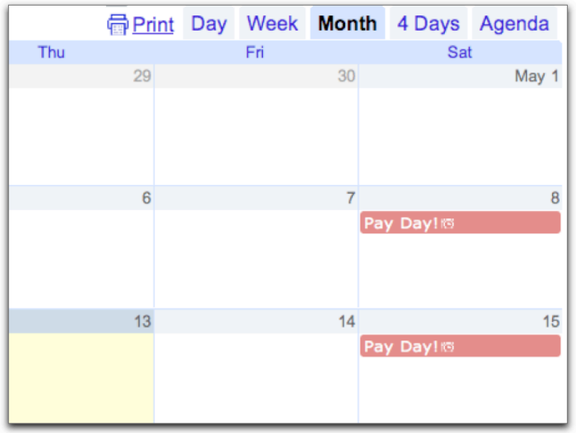
2



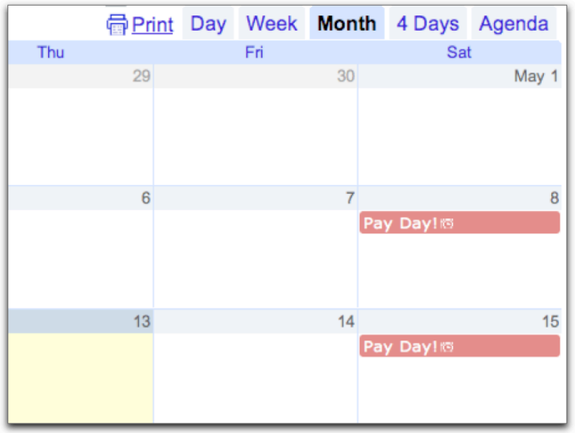
Ryan's Calendar



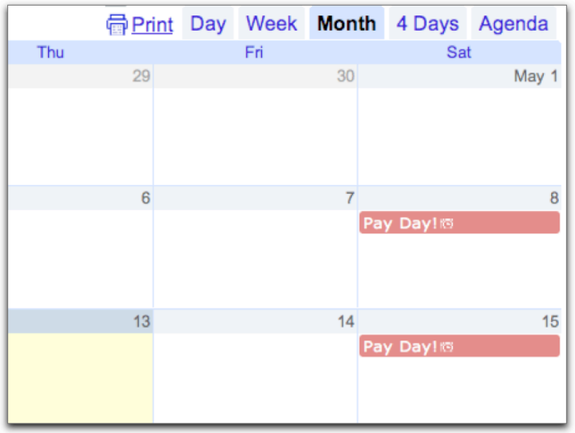
Tim's Calendar



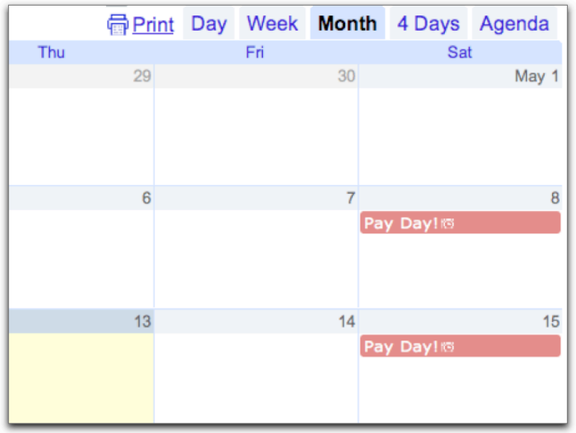
Steve's Calendar



Julia's Calendar



Scott's Calendar



Dan's Calendar

2-Legged OAuth 1 Flow

Registering an Application

0

Developer registers application with Google, gets a **`consumer_key`** and **`consumer_secret`**

Google accounts

Manage payroll.saasyapp.com (Active)

To register your domain, provide the following information. Once you've registered with an authentication certificate, you will be able to use secure tokens when communicating with a Google service.

Target URL path prefix: http://payroll.saasyapp.com/
Must be the prefix of the *next* parameter used in AuthSub.
e.g. http://example.com/authsub

Domain description: (Optional) SaaSy Payroll
Tell us about your domain

OAuth Consumer Key: **payroll.saasyapp.com**

OAuth Consumer Secret: **0/HBCxRybXH+TD7011OypUdM**

We do not have a certificate for your domain.

Upload new X.509 cert: (Optional) Choose File No file chosen
File must be in PEM format. Learn More

Test your AuthSub registration here.

Save Return

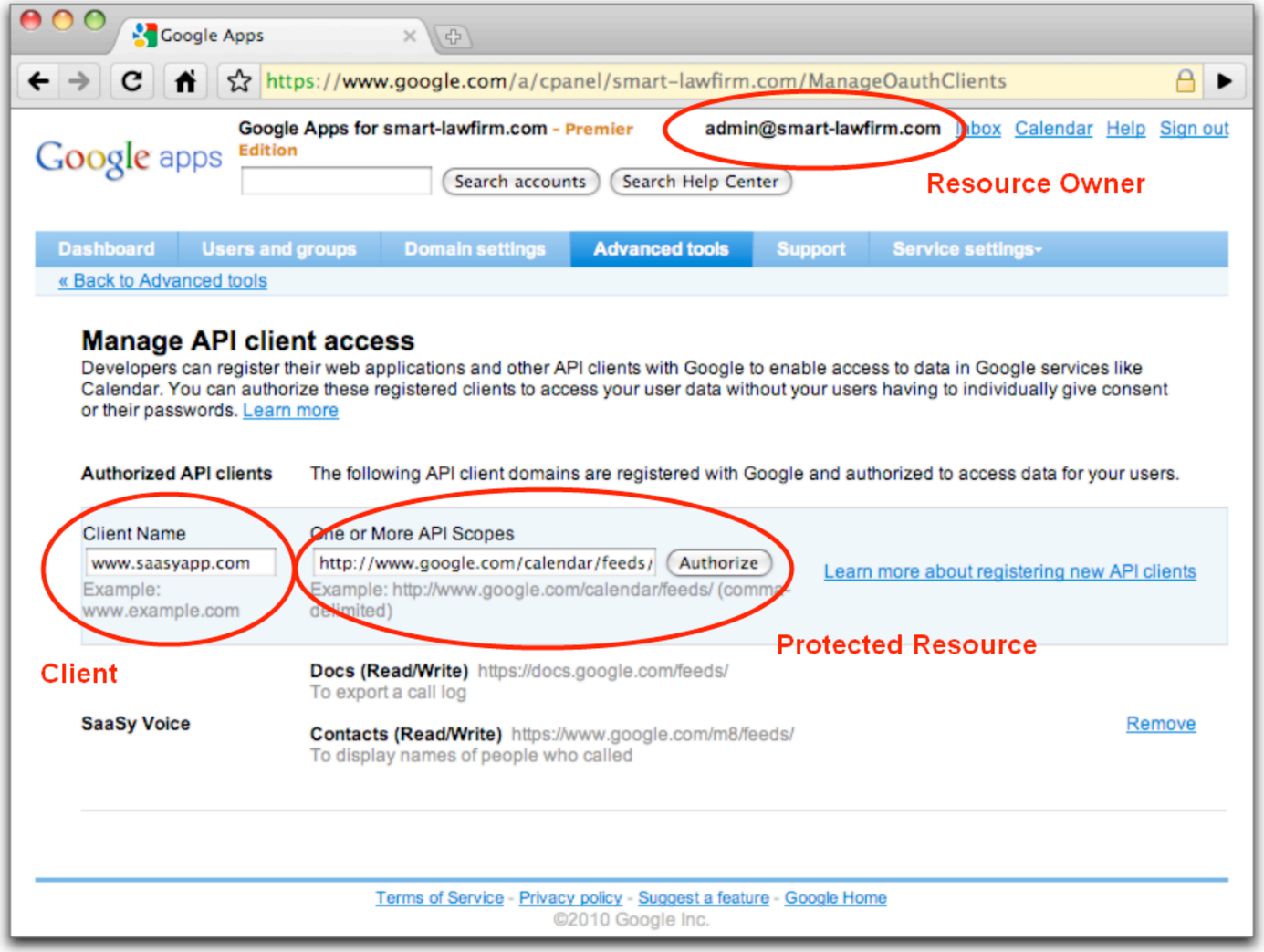
Granting Data Access

1

In an offline process, Google Apps domain administrator grants access to the app, specifying the **`consumer_key`** for the app **`scope`** or APIs the app needs access to

OR

The Google Apps domain administrator approves data access for the app during the installation from the Google Apps Marketplace.



Calling an OAuth Protected API

2

Application makes a HTTP **`GET`** or HTTP **`POST`** request to the server containing the protected resource, including an **`Authorization`** header. Additionally, the application specifies which user's data it is trying to access via a **`xoauth_requestor_id`** query parameter.

```
https://www.google.com/calendar/feeds/default/private
                  /full?xoauth_requestor_id=<email address>
```

Header:

```
Authorization: OAuth
  oauth_version="1.0",
  oauth_nonce="1cbf231409dad9a2341856",
  oauth_timtestamp="123456789",
  oauth_consumer_key="<consumer_key>",
  oauth_signature_method="HMAC-SHA1",
  oauth_signature="1qz%2F%2BfwtsuO"
```

2-legged OAuth 1: The Whole Flow

0 — Developer registers application → Google

1 — Admin visits SaaSy Google, authorizes app → Google

SaaSy Payroll
your payroll solution, your way

2 — SaaSy Payroll accesses Google Calendar using **`xoauth_requestor_id`** and Authorization header with signature →

← Google returns protected data

SaaSy Payroll
your payroll solution, your way

3 — SaaSy Payroll accesses Google Calendar for another user using **`xoauth_requestor_id`** and complicated header with signature →

← Google returns protected data

Recap

Recap

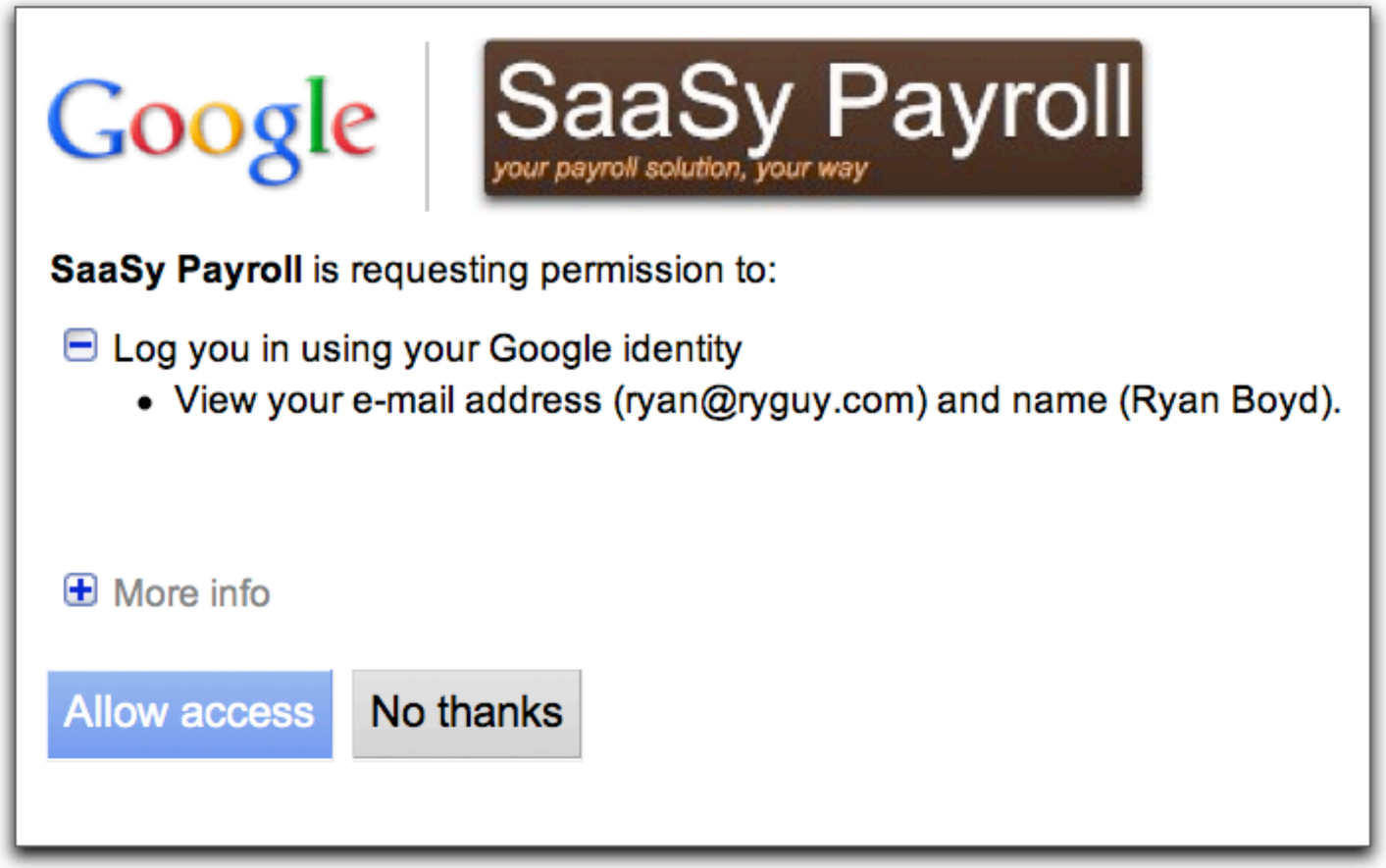
1. Terminology
2. OpenID for Authentication
3. Mobile Authentication
4. OAuth for Authorizing access to data owned by Individuals
5. OAuth for Authorizing access to data owned by Businesses

The Future

One Protocol rules all use cases

Authentication
Authorization

One Protocol Rules all use cases



Resources

Resources

1. Google Identity Toolkit (http://goo.gl/TkkIz). Talk to esachs@google.com for tester access.
2. OAuth Playground for OAuth 1 (http://googlecodesamples.com/oauth_playground/)
3. Google's Auth docs (http://code.google.com/apis/accounts/docs/)
4. ClientLogin #FAIL I/O session from yesterday (http://goo.gl/b78jJ)
5. Ryan's Twitter (@ryguyrg)

Feedback: **http://goo.gl/DpUBh**
#io2011 #TechTalk

Feedback: **http://goo.gl/DpUBh**
#io2011 #TechTalk

Q & A


Google™
11
I/O