OAuth

Muhammad **Rizwan** Asghar

The University of Auckland

September 16, 2014

For template of slides,
thanks to kingsoftstore.com

Overview of OAuth

- An open standard for authorisation
- Inspired by OpenID
- Began in 2006
- OAuth 1.0 was published as RFC 5849 in April 2010

Why OAuth

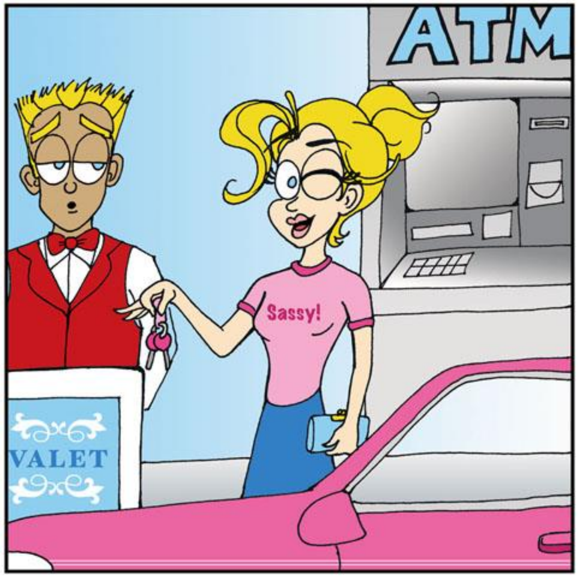
- An alternate to password sharing

Source: mizwhiz.com

Basic Purpose

- Authorisation
- Without password sharing

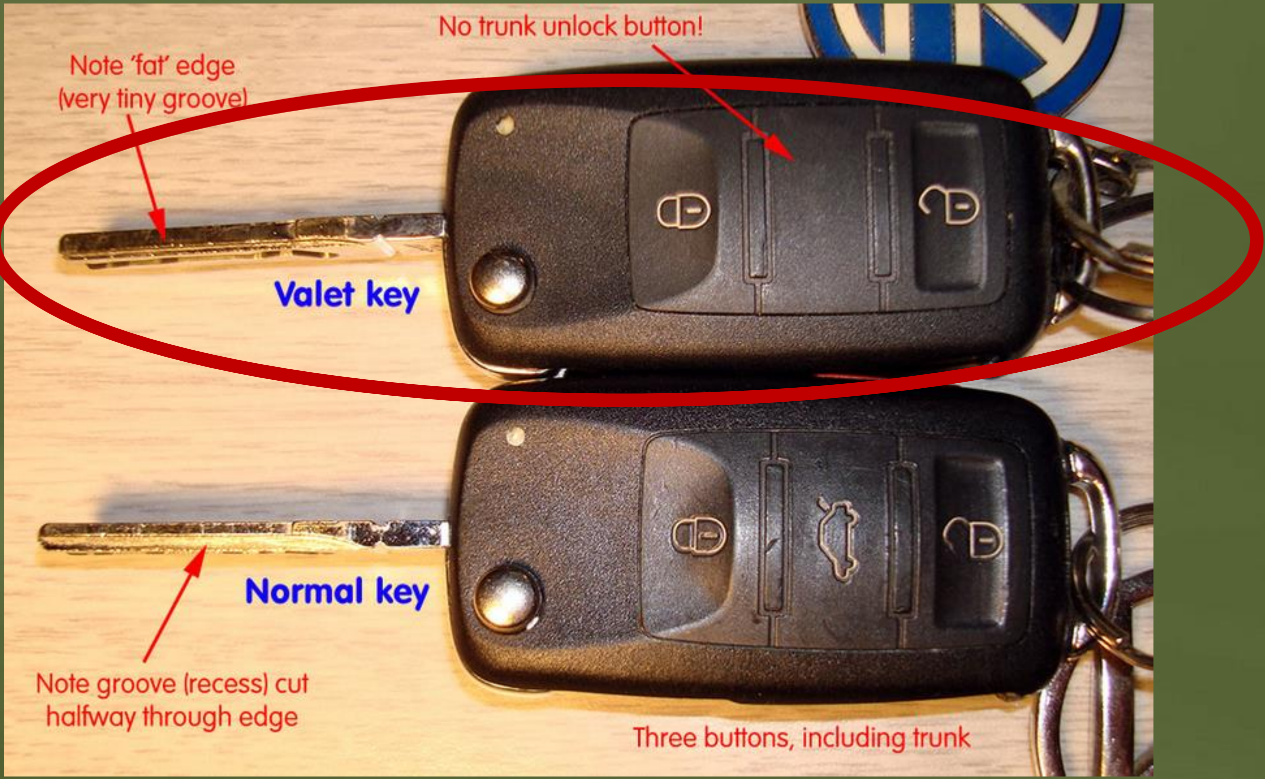
Valet Parking



"If God intended us to walk,
He wouldn't have invented Valet Parking!"

Source: kerripom.com

Can we Limit Access?



Source: photobucket.com

In Our Use-Case Scenario

- Car is like a *Server*
- Sassy is a Resource Owner
- Parking attendant is a Client
- Normal key is like *Credentials*
- Valet key is a Token

In OAuth

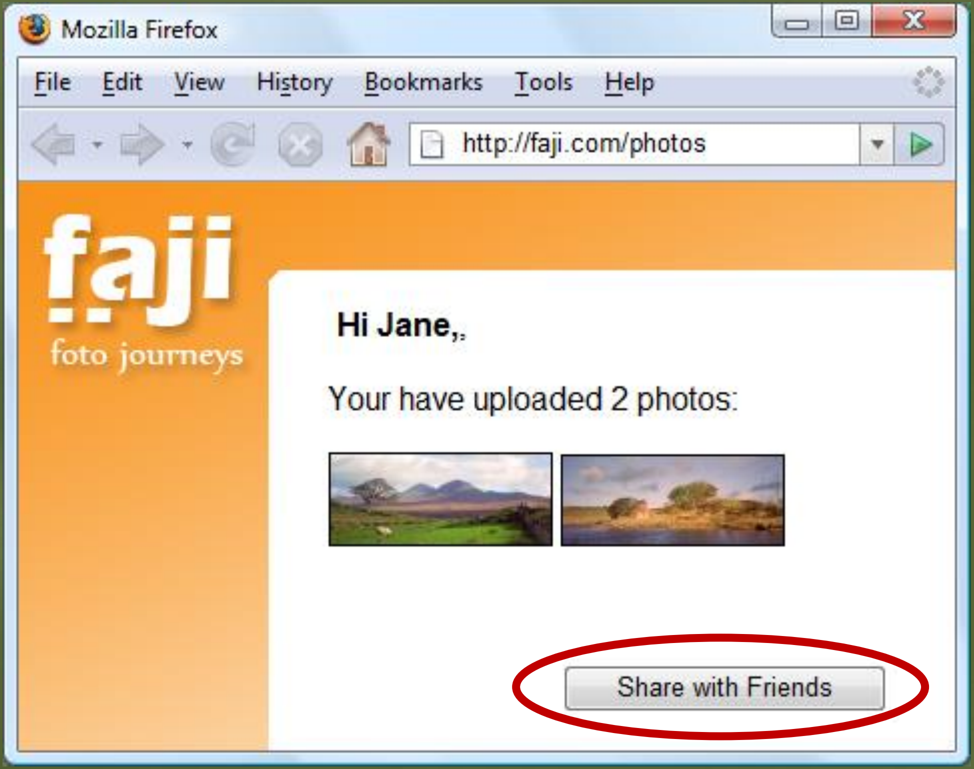
- Resource Owner grants access to protected Resources hosted by Servers
- The Client gets Tokens
- For gaining access, a Client presents Tokens to the Server

Example

- Jane (Resource Owner) is back from her Scotland vacation
- For sharing journey photos with friends, Jane uses Faji (Server), a photo sharing site
- She signs into her faji.com account, and uploads two photos

Source: hueniverse.com/oauth

1 – Photo Sharing



Hi Jane,,

Your have uploaded 2 photos:

Source: hueniverse.com/oauth

Example Cont.

- Jane wants to print photos for grandmother who does not have the Internet connection
- Jane uses Beppa, an environment friendly photo printing service
- Jane visits beppa.com to order prints

Source: hueniverse.com/oauth

Naïve Approach

- Jane can provide her Faji credentials (including password) to Beppa
- There are serious issues with password sharing

Limitations of Password Sharing

- Lack of trust
- No support for granular permissions, thus violating the principle of least privileges
- Once granted, no possibility to revoke access (unless password is changed)

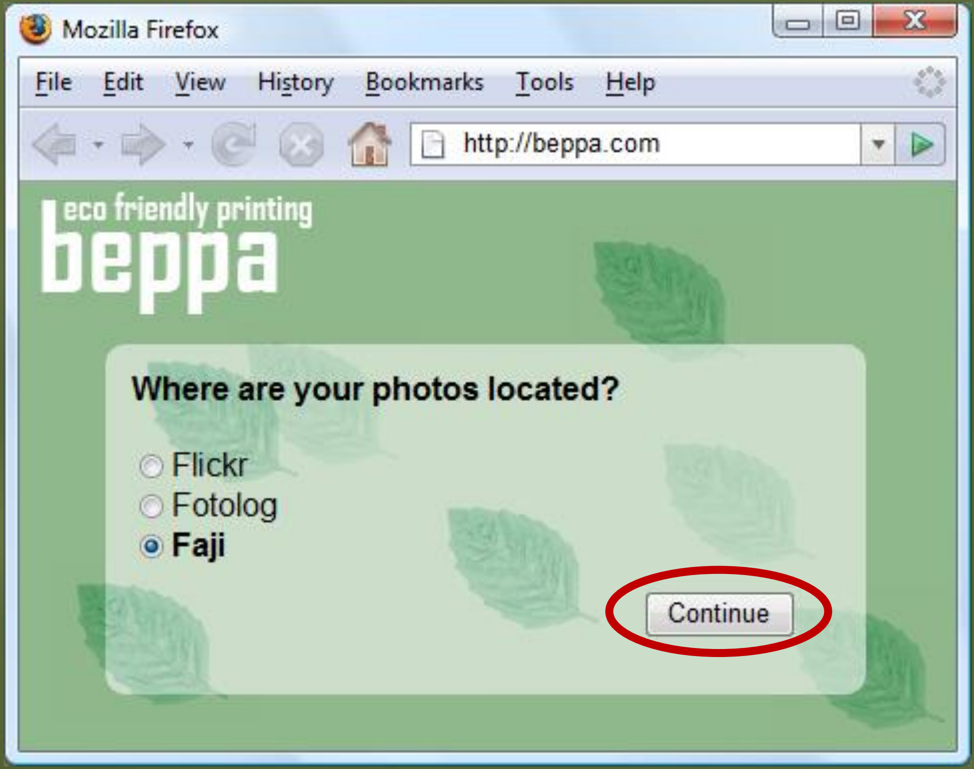
What is OAuth?

- OAuth enables access delegation
  - To selected Clients
  - For certain Resources hosted by Servers
  - For limited time
  - With possibility of revocation

Proposed Approach

- Using OAuth, Jane can grant Beppa access to her photos without password sharing
- Beppa can contact Faji to request access to the photos

2 – Choose Server



http://beppa.com

Where are your photos located?

○ Flickr
○ Fotolog
● Faji

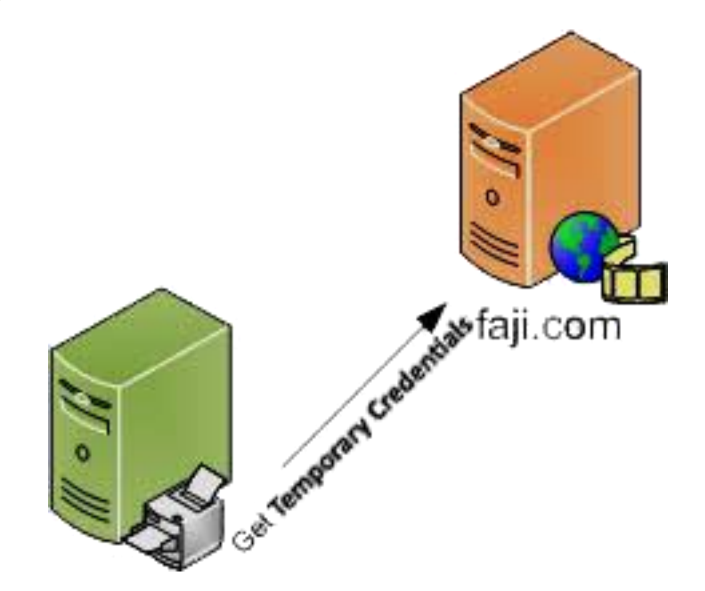
Source: hueniverse.com/oauth

Example Cont.

- When Beppa added support for Faji photo import, the Beppa developer obtained a set of Client Credentials from Faji
- Client Credentials
  - Client Identifier
  - Secret

Source: hueniverse.com/oauth

3 – Temporary Credentials



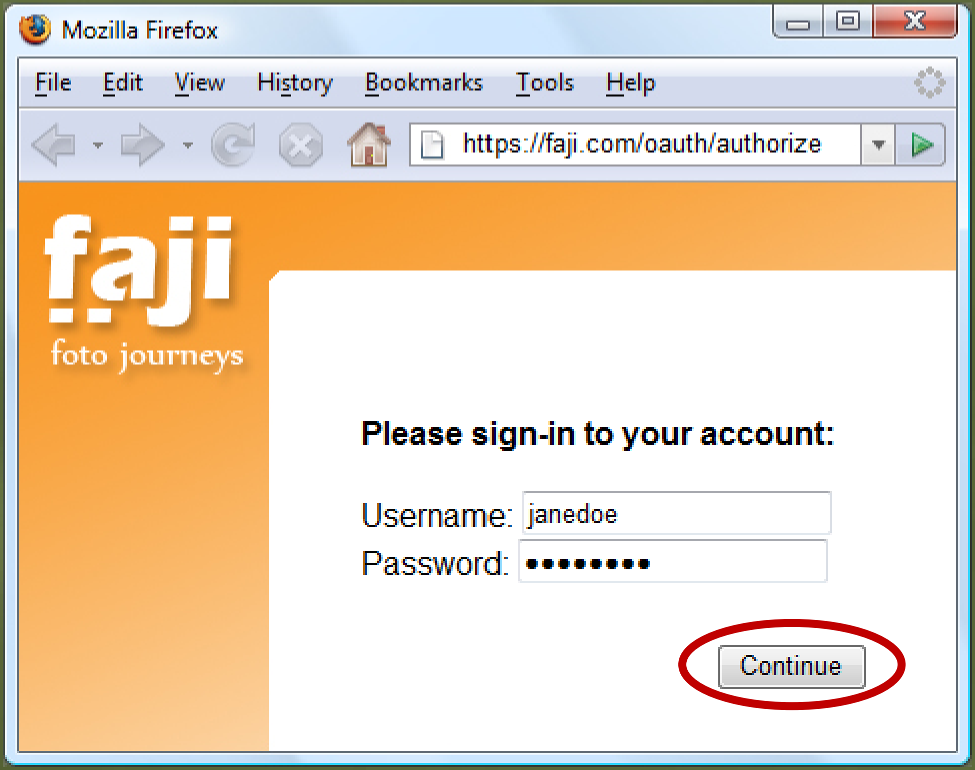
Source: hueniverse.com/oauth

Example Cont.

- Temporary Credentials
  - Request Token
  - …
- After receiving Temporary Credentials, Beppa redirects Jane to Faji (with Temporary Credentials)

Source: hueniverse.com/oauth

4 – Signing-in to Faji



Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://faji.com/oauth/authorize
faji
foto journeys

Please sign-in to your account:

Username: janedoe

Password: •••••••••

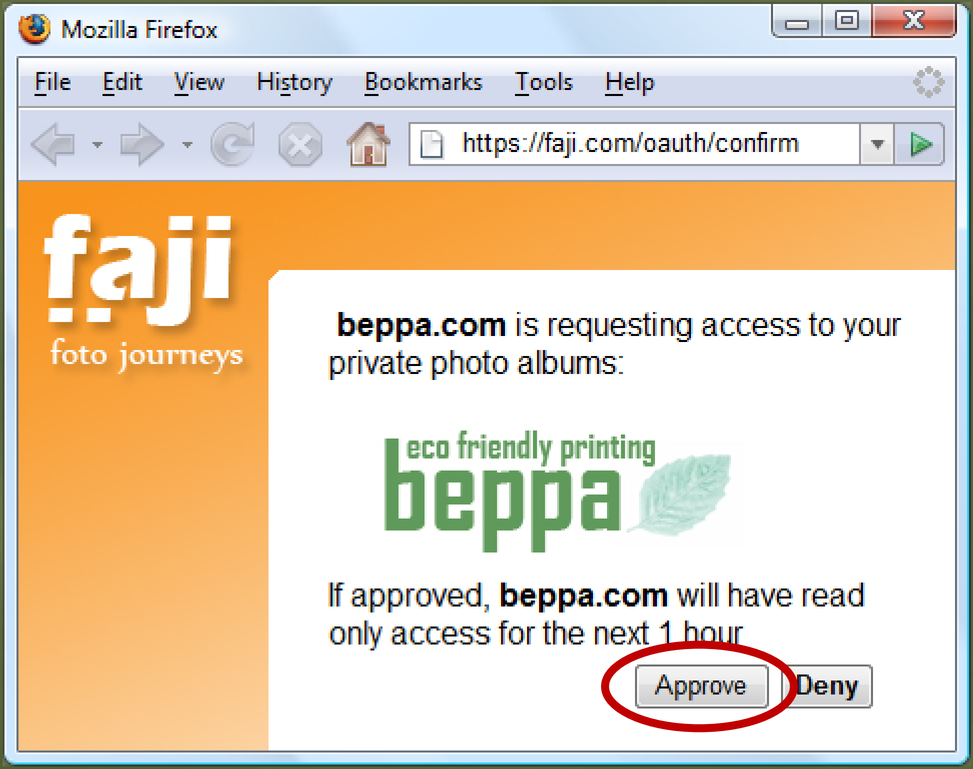
Source: hueniverse.com/oauth

Example Cont.

- Jane provides her username and password only to Faji
- Once Jane logs in to Faji, she is asked to approve or deny request from Beppa
- The request includes scope and lifetime

Source: hueniverse.com/oauth

5 – Authorisation Approval



Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://faji.com/oauth/confirm
faji
foto journeys

beppa.com is requesting access to your private photo albums:

eco friendly printing
beppa

If approved, **beppa.com** will have read only access for the next 1 hour

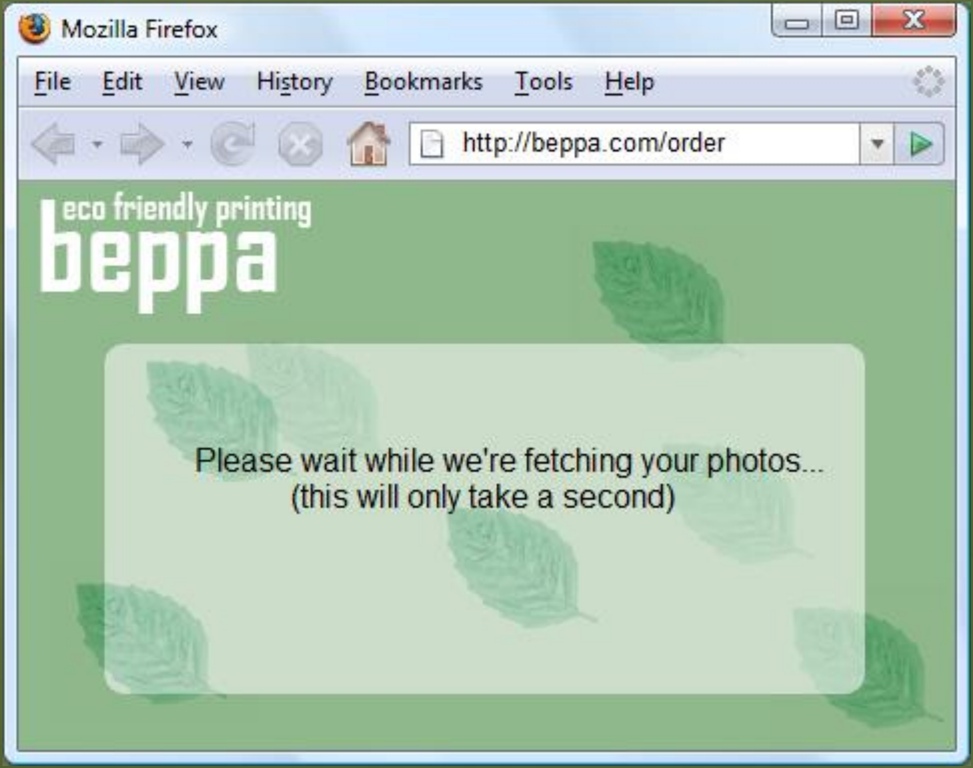
Source: hueniverse.com/oauth

Example Cont.

- After approval, Faji marks the Temporary Credentials as resource-owner-authorised
- Jane is redirected back to Beppa together with the Temporary Credentials Identifier
- This allows Beppa to fetch photos

Source: hueniverse.com/oauth

Requesting Photos



Please wait while we're fetching your photos...
(this will only take a second)

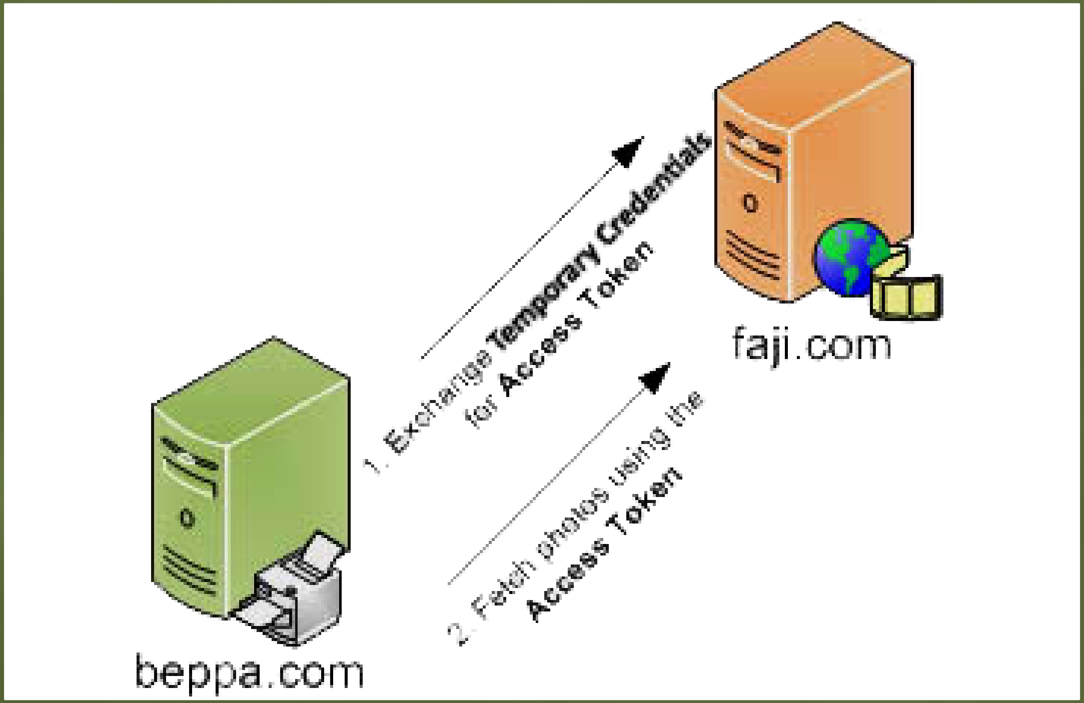
Source: hueniverse.com/oauth

Example Cont.

- While Jane waits, Beppa uses the authorised Temporary Credential and exchanges it for an Access Token
- Temporary Credentials are only good for obtaining User approval
- Access Tokens are used to access Resources, i.e., photos

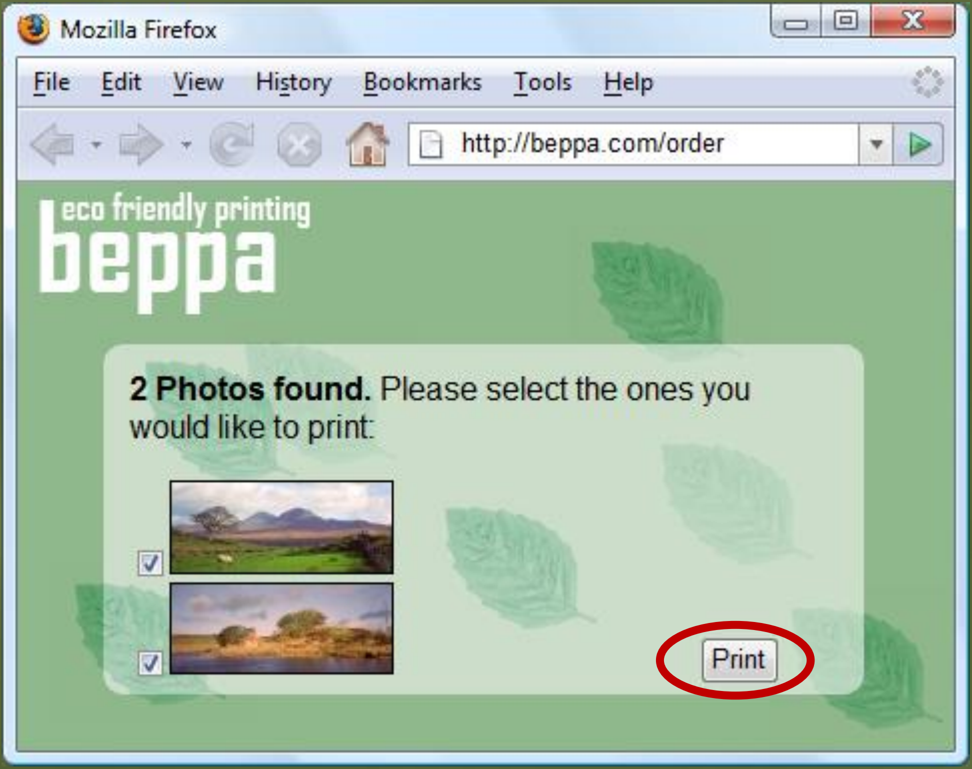
Source: hueniverse.com/oauth

6 – Access Tokens


faji.com
1. Exchange Temporary Credentials for Access Token
2. Fetch photos using the Access Token
beppa.com

Source: hueniverse.com/oauth

7 – Fetched Photos



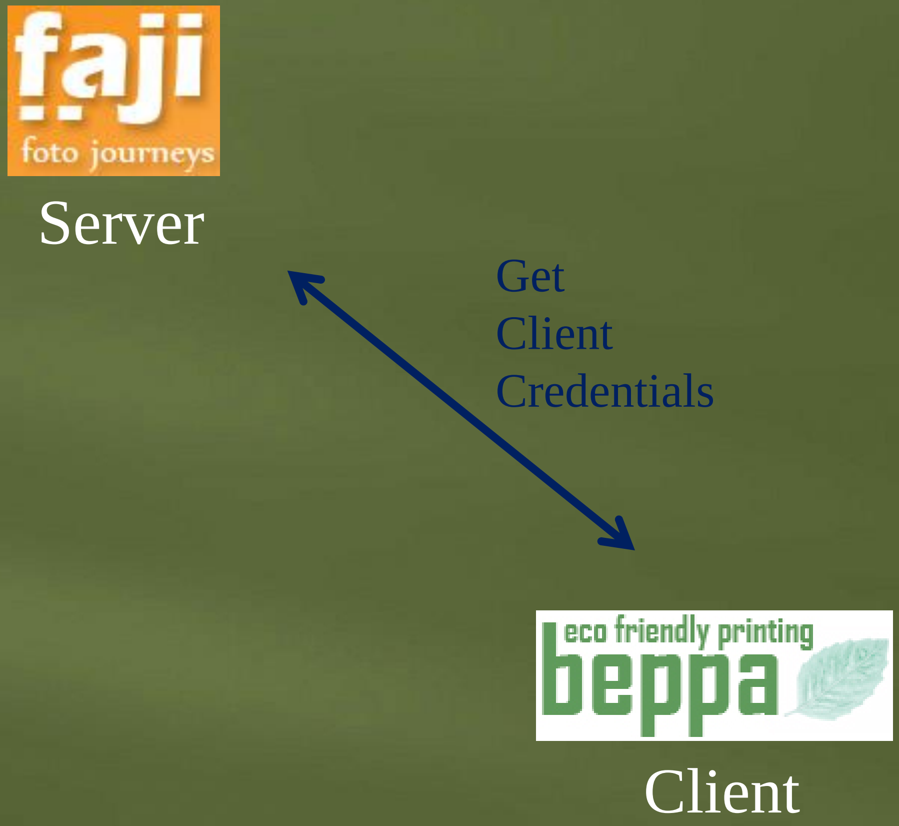
2 Photos found. Please select the ones you would like to print:

☑

☑

Source: hueniverse.com/oauth

OAuth Workflow: Application Registration

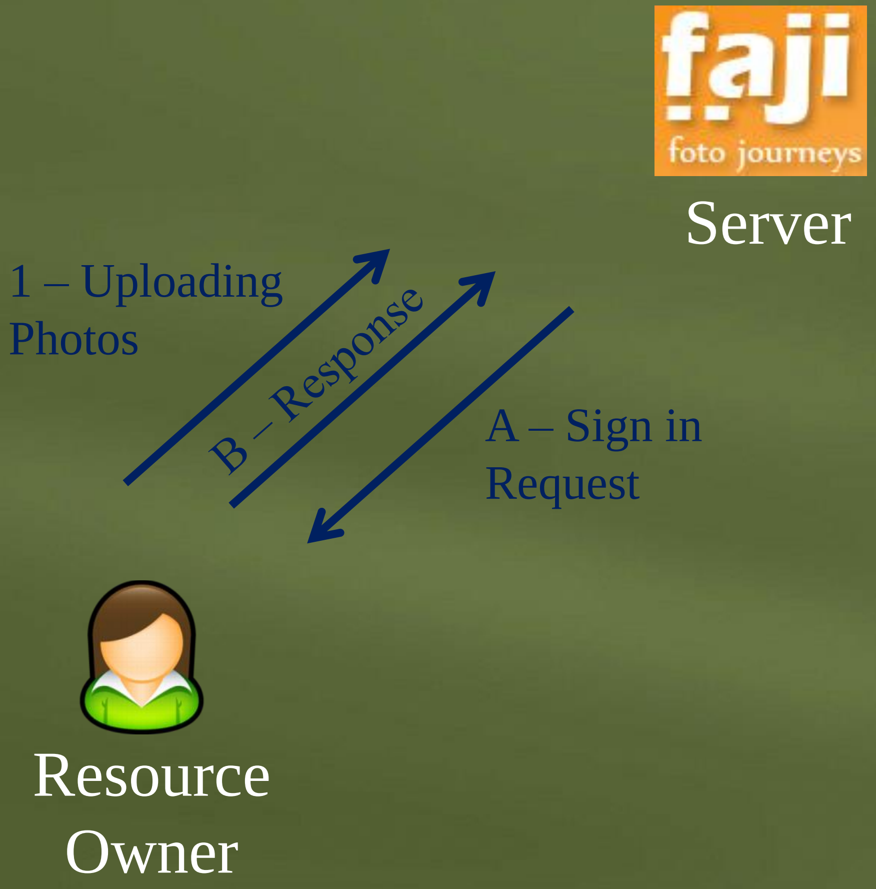


Server

Get
Client
Credentials

Client

OAuth Workflow: Managing Resources



faji foto journeys

Server

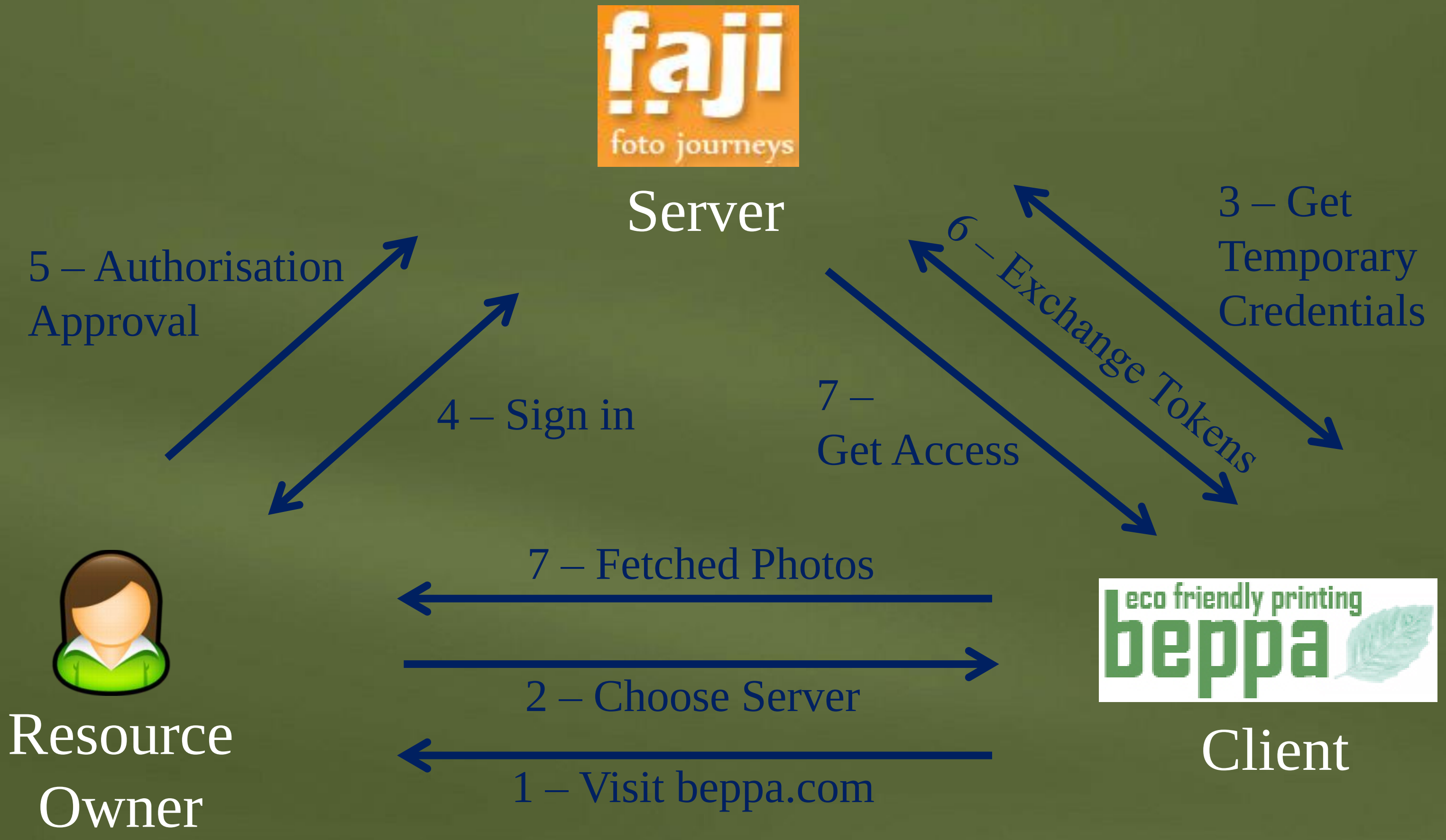
1 – Uploading Photos

B – Response

A – Sign in Request

Resource Owner

OAuth Workflow



Server

5 – Authorisation Approval

4 – Sign in

7 – Get Access

6 – Exchange Tokens

3 – Get Temporary Credentials

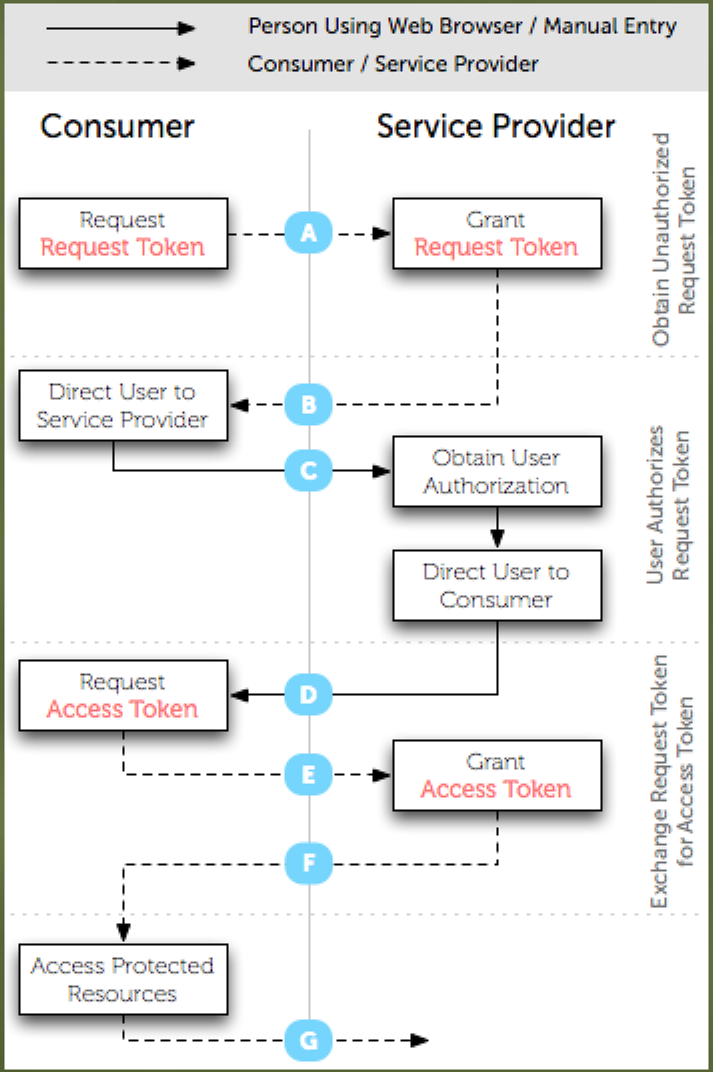
7 – Fetched Photos

2 – Choose Server

1 – Visit beppa.com

Resource Owner

Client



Person Using Web Browser / Manual Entry

Consumer / Service Provider

Consumer

Service Provider

Obtain Unauthorized Request Token

Request Request Token

A

Grant Request Token

User Authorizes Request Token

Direct User to Service Provider

B

C

Obtain User Authorization

Direct User to Consumer

Exchange Request Token for Access Token

Request Access Token

D

E

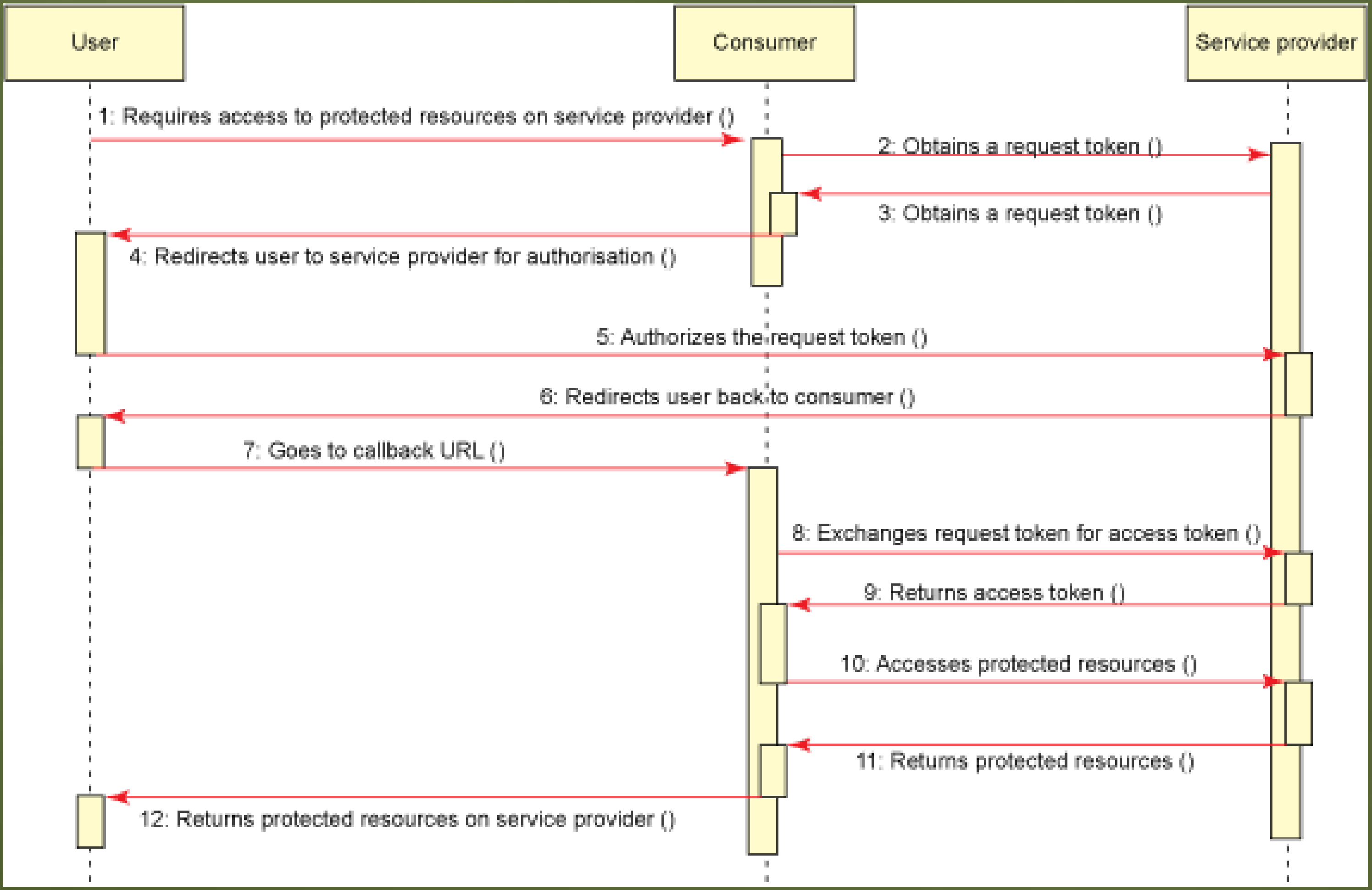
Grant Access Token

F

Access Protected Resources

G

Source: http://oauth.net/core/1.0


User
Consumer
Service provider
1: Requires access to protected resources on service provider ()
2: Obtains a request token ()
3: Obtains a request token ()
4: Redirects user to service provider for authorisation ()
5: Authorizes the request token ()
6: Redirects user back to consumer ()
7: Goes to callback URL ()
8: Exchanges request token for access token ()
9: Returns access token ()
10: Accesses protected resources ()
11: Returns protected resources ()
12: Returns protected resources on service provider ()

Source: ibm.com

OAuth Details

- Clients register with the Server and obtain Client Credentials
- Clients obtain Temporary Credentials from the Server
- For gaining access to Resources, Clients get Token Credentials
  - Access Token
  - …

OAuth Details (2)

- After approval by Resource Owner, a Client can exchange Request Tokens for Access Tokens
- Using Access Tokens, a Client can get access to protected Resources

Request Token vs Access Token

- Request Token
  - A value used by the Client to obtain authorisation from the Resource Owner
- Access Token
  - A value used by the Client to gain access to the Protected Resources on behalf of the Resource Owner

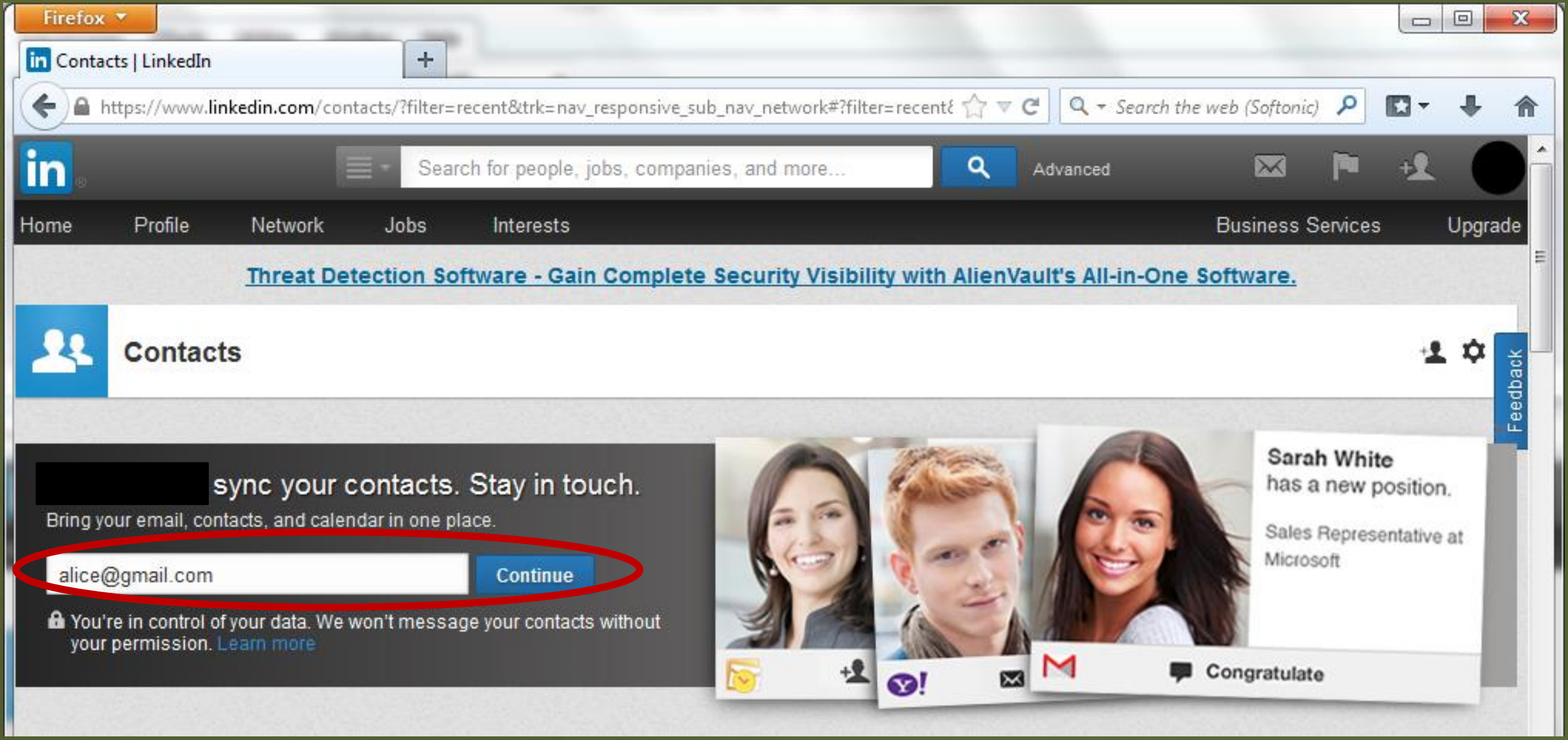
Request Token vs Access Token (2)

- Request Token is also known to Resource Owner
- Access Token is known to the Client
- Request Tokens have a limited lifetime
- Access Tokens can be larger in size

Other Use-Case Scenarios

- LinkedIn accessing GMail contacts
  - LinkedIn is a Client
  - GMail is Server
- Third-party applications accessing Tweets
  - Third-party is a Client
  - Twitter is a Server

LinkedIn Accessing Gmail Contacts


Firefox
Contacts | LinkedIn
https://www.linkedin.com/contacts/?filter=recent&trk=nav_responsive_sub_nav_network#?filter=recent&
Search the web (Softonic)
Search for people, jobs, companies, and more...
Advanced
Home
Profile
Network
Jobs
Interests
Business Services
Upgrade
Feedback
Sarah White
has a new position.
Sales Representative at
Microsoft
Congratulate

Threat Detection Software - Gain Complete Security Visibility with AlienVault's All-in-One Software.

Contacts

[redacted] sync your contacts. Stay in touch.

Bring your email, contacts, and calendar in one place.

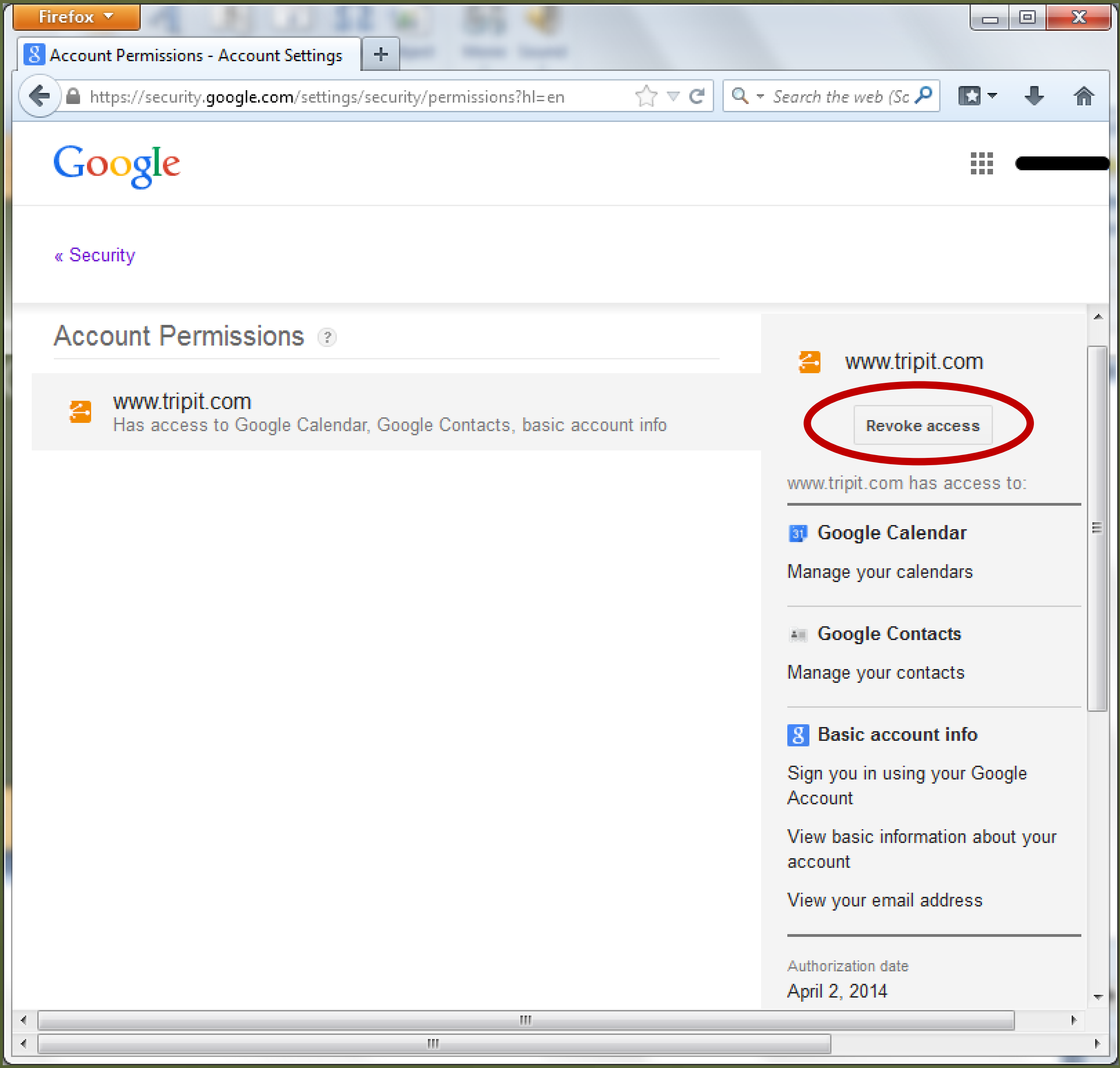
alice@gmail.com

Continue

You're in control of your data. We won't message your contacts without your permission. Learn more

Access Management

- GMail (Server)
  - Settings
    - Accounts and Import
      - Other Google Account Settings
        - Security
        - Account Permissions (View All)



Firefox
Account Permissions - Account Settings
https://security.google.com/settings/security/permissions?hl=en
Search the web (Sc
Google

[redacted]

« Security

Account Permissions ?

www.tripit.com
Has access to Google Calendar, Google Contacts, basic account info

www.tripit.com

Revoke access

www.tripit.com has access to:

Google Calendar
Manage your calendars

Google Contacts
Manage your contacts

Basic account info
Sign you in using your Google Account

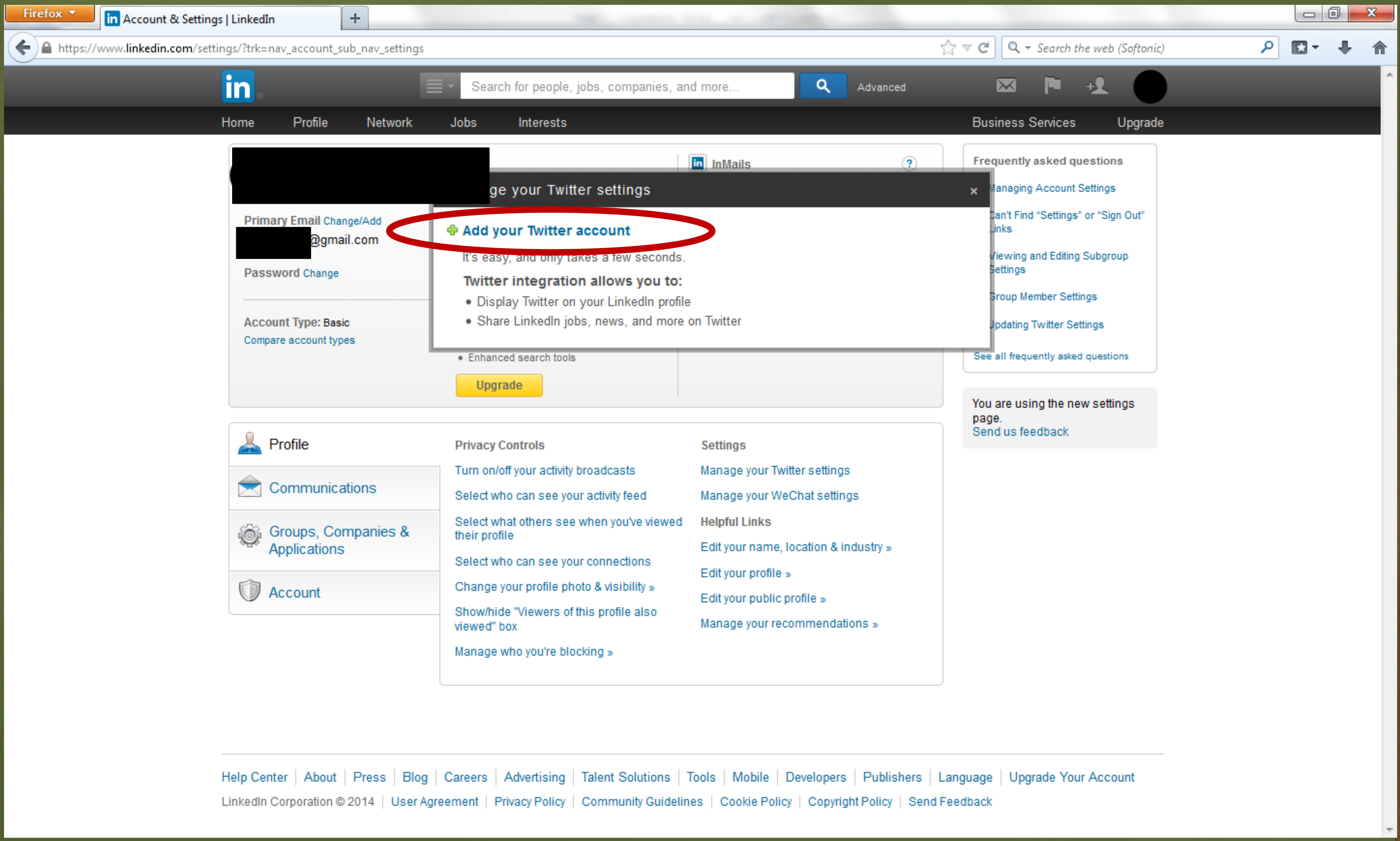
View basic information about your account

View your email address

Authorization date
April 2, 2014

Twitter Integration

- LinkedIn (Client)
  - Account & Settings
    - Privacy & Settings
      - Profile
        - Settings
        - Manage your Twitter settings


Firefox
Account & Settings | LinkedIn
https://www.linkedin.com/settings/?trk=nav_account_sub_nav_settings
Search the web (Softonic)
in
Search for people, jobs, companies, and more...
Advanced
Home
Profile
Network
Jobs
Interests
Business Services
Upgrade

[redacted]

Primary Email Change/Add

[redacted]@gmail.com

Password Change

Account Type: Basic

Compare account types

InMails

• Enhanced search tools

Upgrade

ge your Twitter settings ×

Add your Twitter account

It's easy, and only takes a few seconds.

Twitter integration allows you to:

- Display Twitter on your LinkedIn profile
- Share LinkedIn jobs, news, and more on Twitter

Frequently asked questions

lanaging Account Settings

Can't Find "Settings" or "Sign Out" Links

Viewing and Editing Subgroup Settings

Group Member Settings

Updating Twitter Settings

See all frequently asked questions

You are using the new settings page.

Send us feedback

Profile

Communications

Groups, Companies & Applications

Account

Privacy Controls

Turn on/off your activity broadcasts

Select who can see your activity feed

Select what others see when you've viewed their profile

Select who can see your connections

Change your profile photo & visibility »

Show/hide "Viewers of this profile also viewed" box

Manage who you're blocking »

Settings

Manage your Twitter settings

Manage your WeChat settings

Helpful Links

Edit your name, location & industry »

Edit your profile »

Edit your public profile »

Manage your recommendations »

Help Center | About | Press | Blog | Careers | Advertising | Talent Solutions | Tools | Mobile | Developers | Publishers | Language | Upgrade Your Account

LinkedIn Corporation © 2014 | User Agreement | Privacy Policy | Community Guidelines | Cookie Policy | Copyright Policy | Send Feedback

OAuth Limitations

- Privacy issues
  - Servers will know more about Resource Owners and Clients
- Denial-of-Service (DoS) attack
  - Effect on Clients and Servers

Confidentiality

- No guarantee of confidentiality
  - Request
  - Content
- OAuth suggests Servers to protect sensitive resources by employing Transport-Layer Security (TLS)

Integrity using Digital Signatures

- To ensure integrity of requests, OAuth offers three methods
  - PLAINTEXT
    - To be used only with HTTPS
  - HMAC
    - Shared secret
  - RSA
    - Client holds a signing key and Server holds a verification key

Replay Attack

- Vulnerable to replay attack
- To withstand replay attack, OAuth uses:
  - Nonce and
  - Timestamp

Phishing Attack

- OAuth requires redirection to Servers
- Attackers may steal password of careless Resource Owners
- It is up to Resource Owners to verify the authenticity of these websites before entering their credentials

Summary

- OAuth is inspired by OpenID
- It enables access delegation without sharing passwords
- Two layers of tokens provide usability as well as security features
- Many websites offer OAuth-enabled APIs

Q&A

References

- OAuth, http://oauth.net/
- The OAuth 1.0 Protocol, http://tools.ietf.org/html/rfc5849
- OAuth Specifications, http://oauth.net/documentation/spec/
- Authentication, http://nouncer.com/oauth/signature-rfc.html
- OAuth Security Cheatsheet, http://www.oauthsecurity.com/