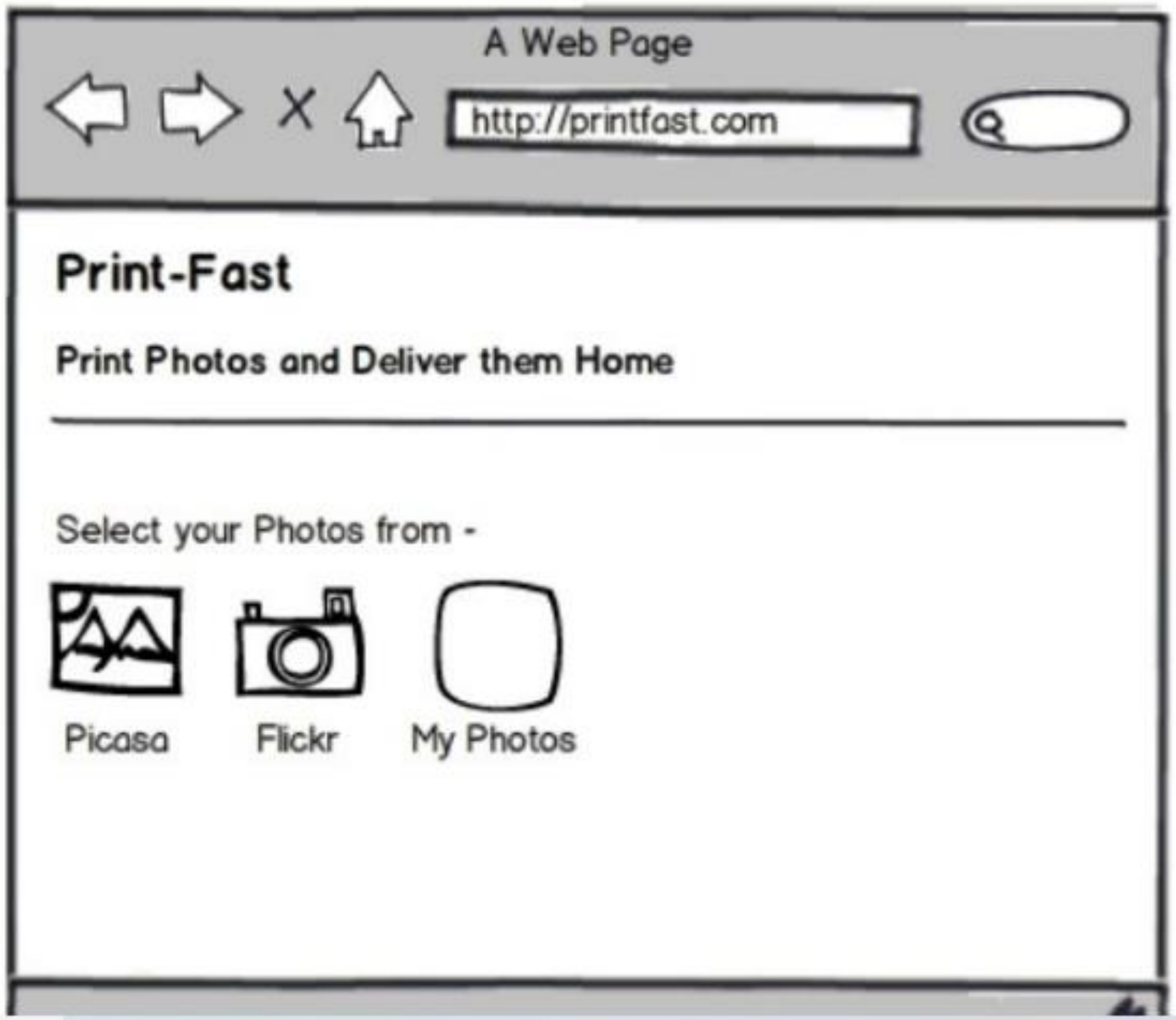
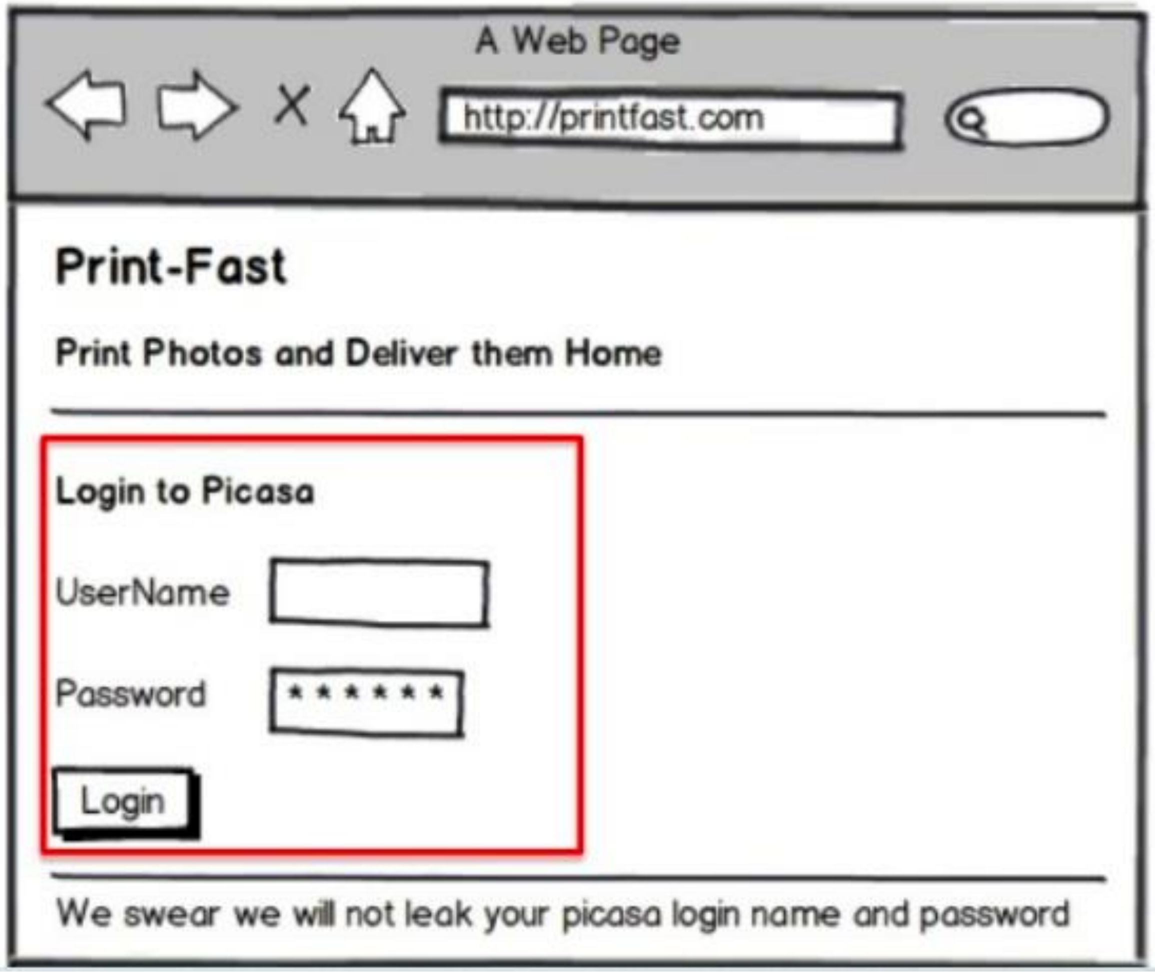
OAUTH

Without OAuth



Without OAuth



A Web Page

http://printfast.com

Print-Fast

Print Photos and Deliver them Home

Login to Picasa

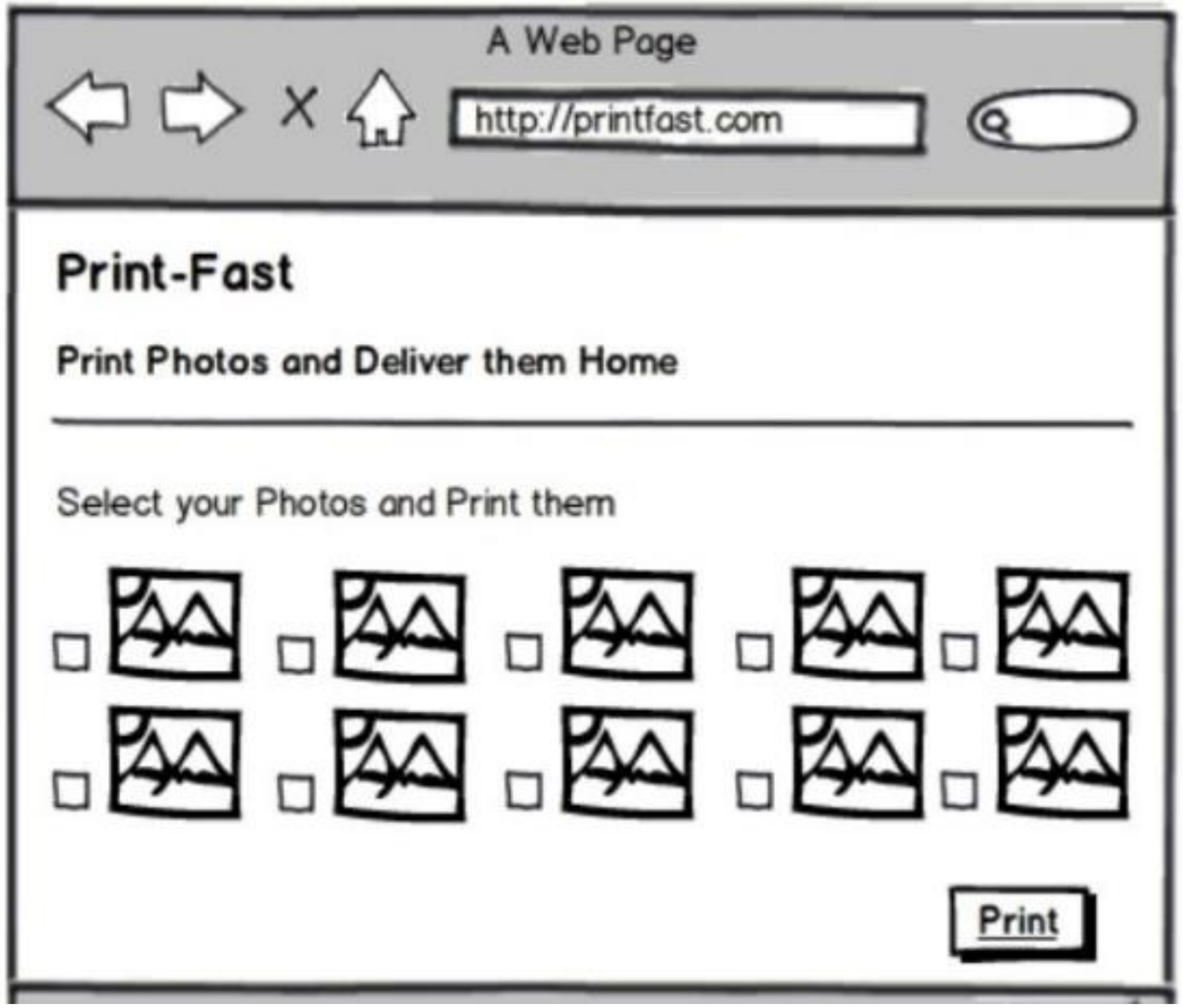
UserName

Password ******

Login

We swear we will not leak your picasa login name and password

Without OAuth


A Web Page

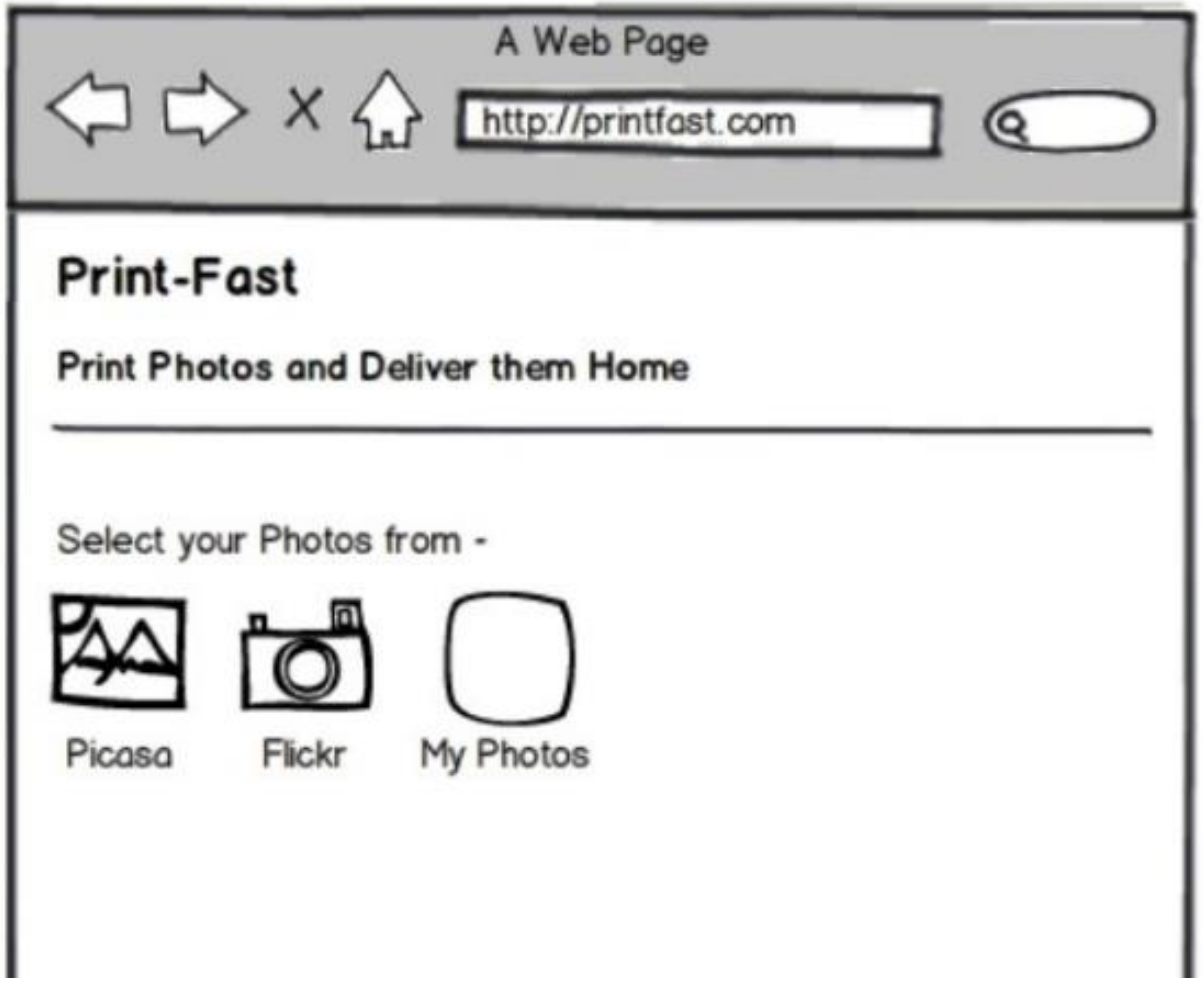
http://printfast.com

Print-Fast

Print Photos and Deliver them Home

Select your Photos and Print them

With OAuth



A Web Page

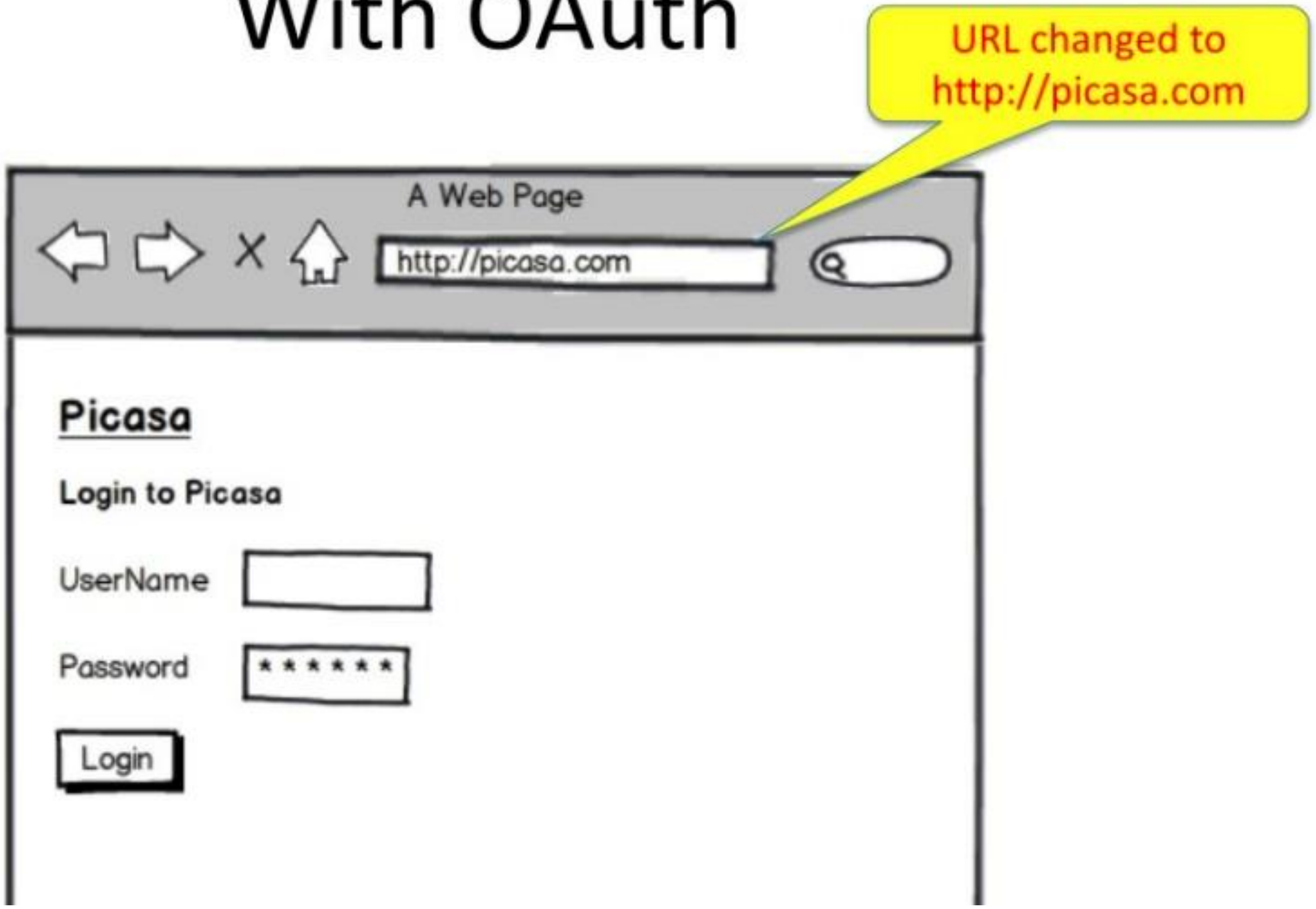
http://printfast.com

Print-Fast

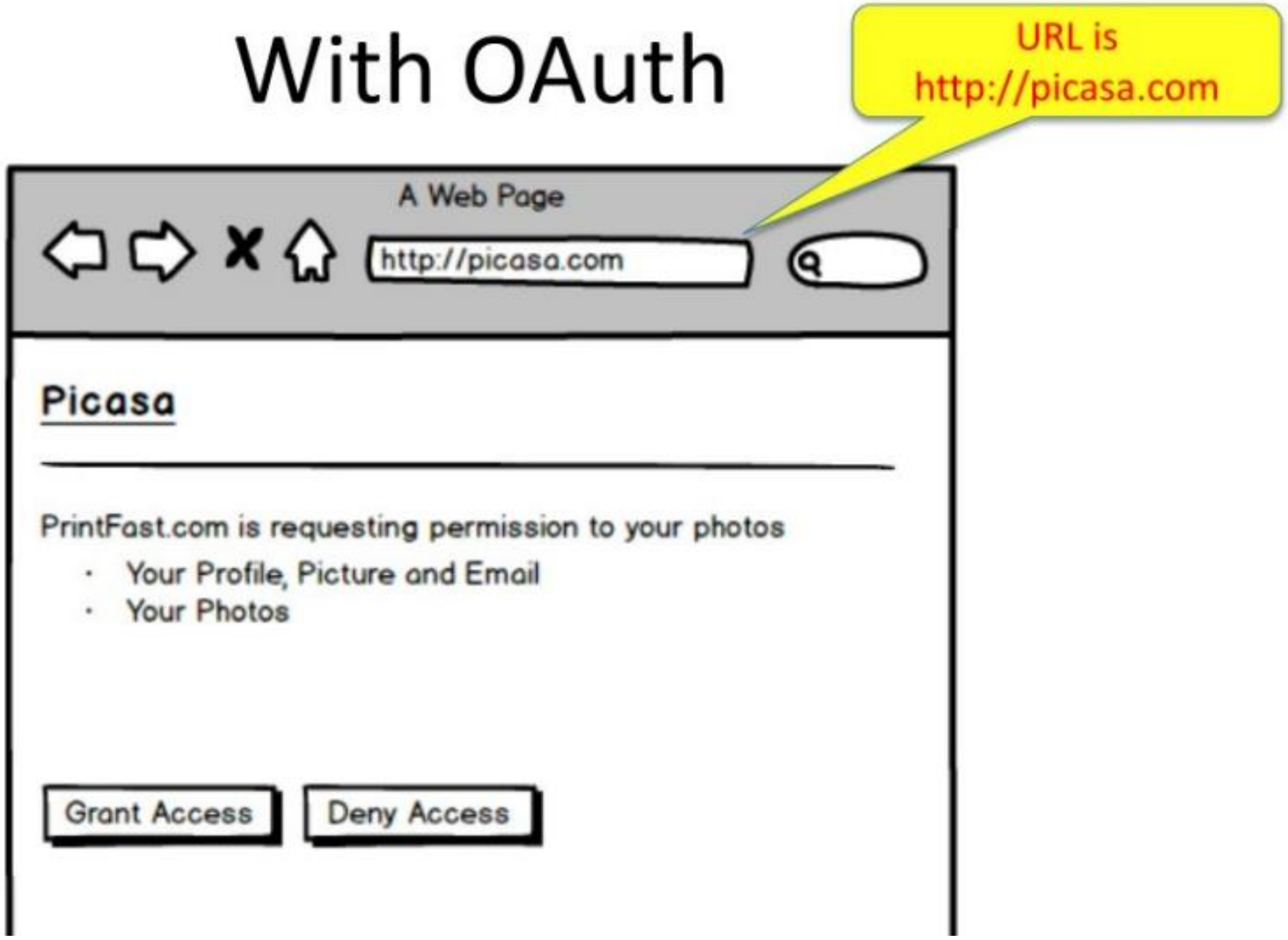
Print Photos and Deliver them Home

Select your Photos from -

With OAuth


URL changed to http://picasa.com
A Web Page
http://picasa.com
Picasa
Login to Picasa
UserName
Password
******
Login

With OAuth

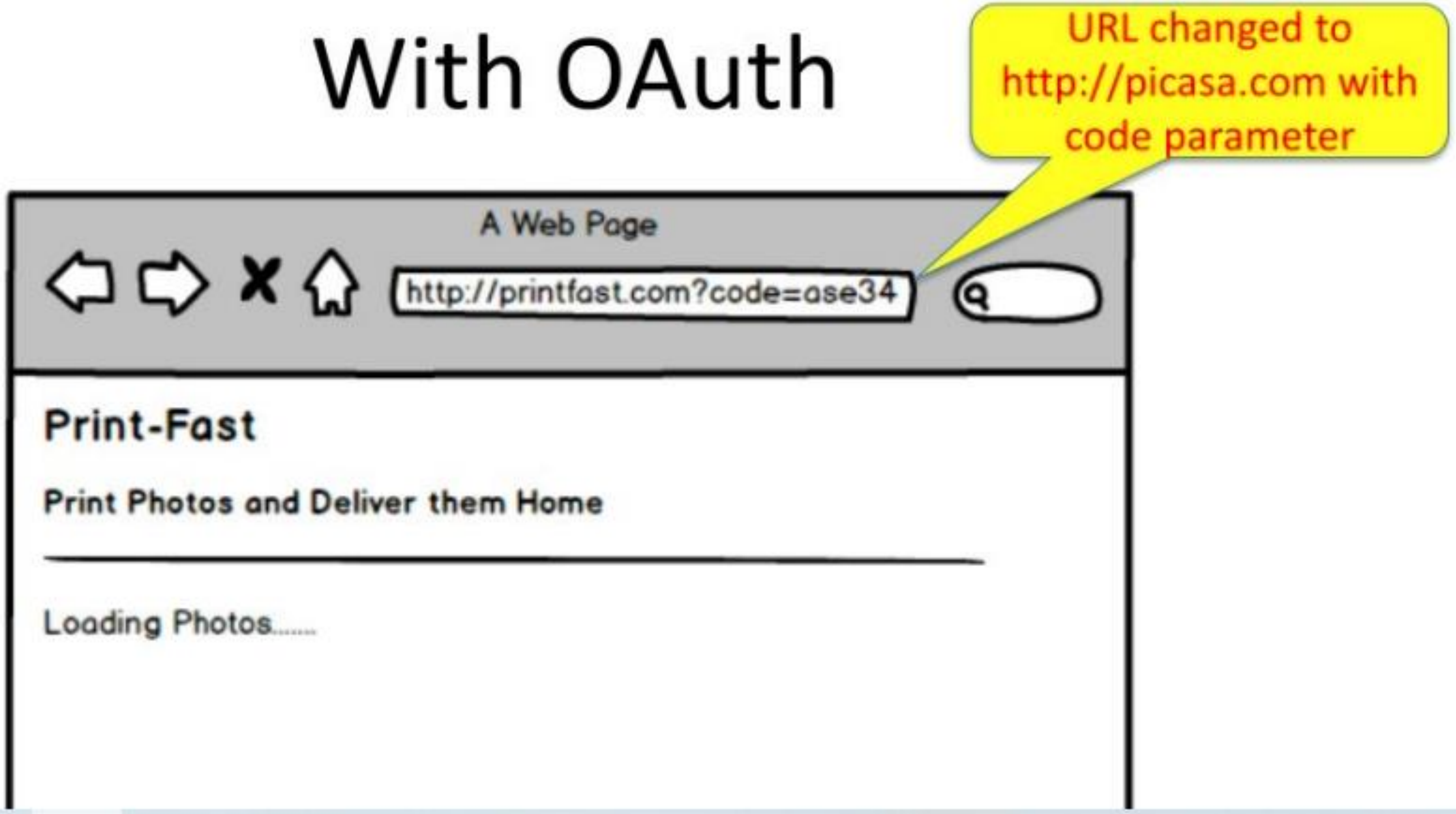


Picasa

PrintFast.com is requesting permission to your photos

- Your Profile, Picture and Email
- Your Photos

With OAuth

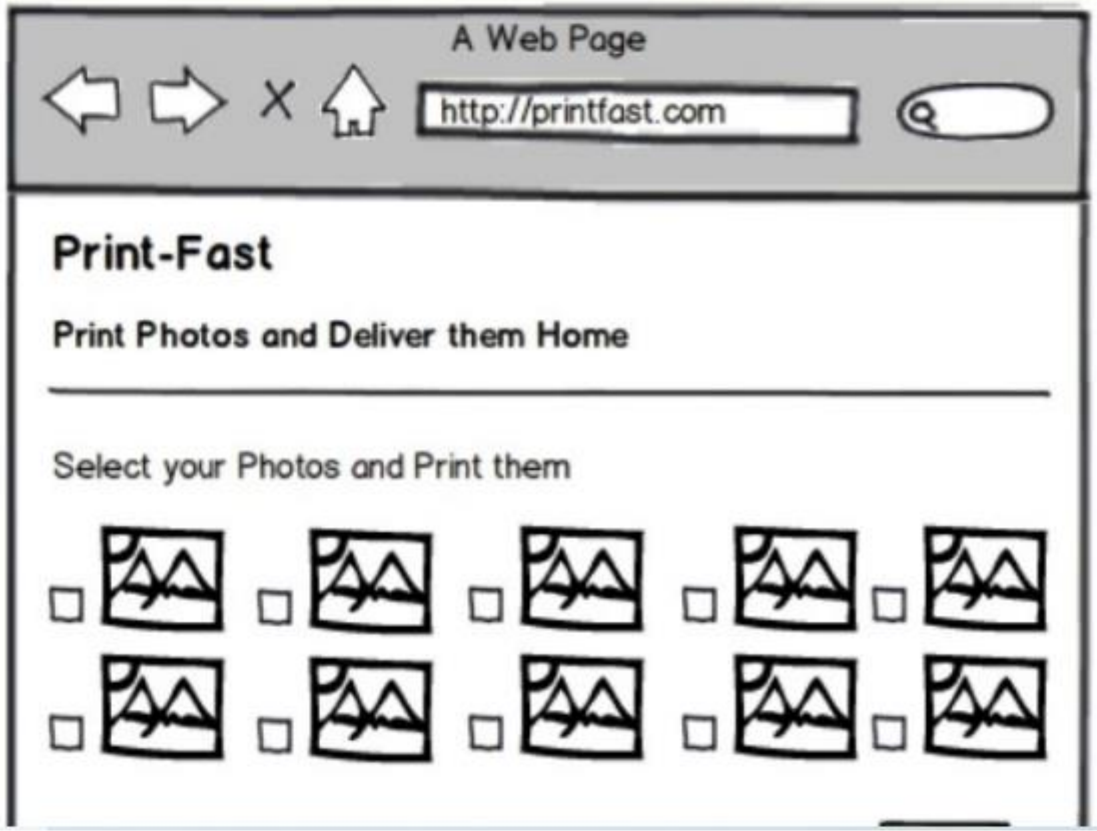

URL changed to http://picasa.com with code parameter
A Web Page
http://printfast.com?code=ase34

Print-Fast

Print Photos and Deliver them Home

Loading Photos.......

With OAuth

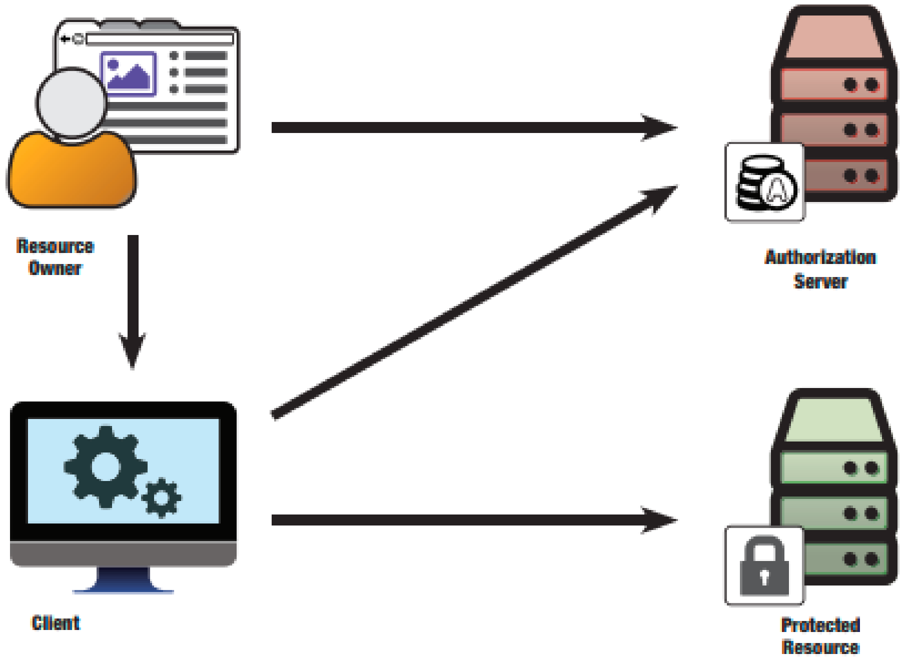


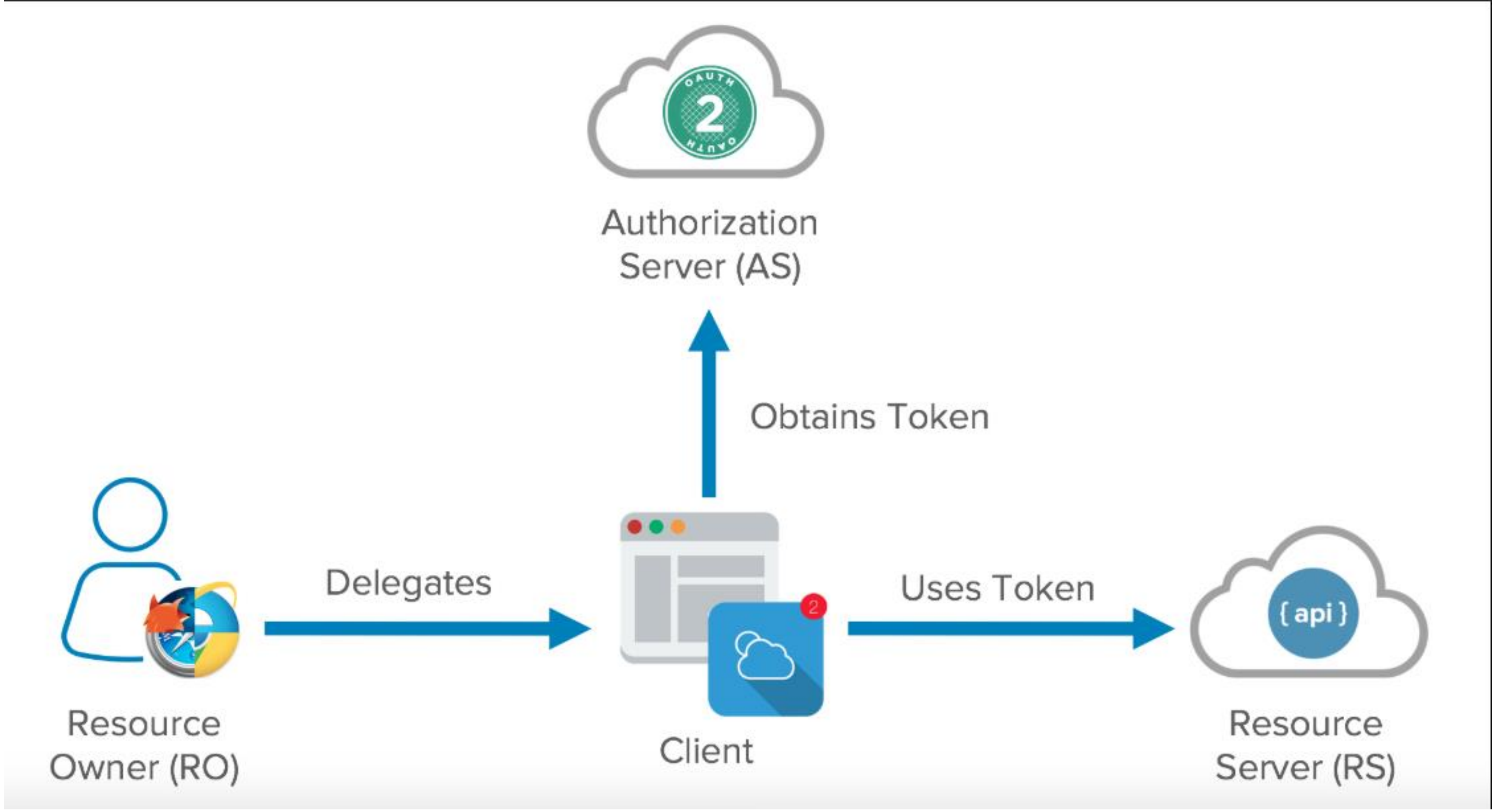
Print-Fast

Print Photos and Deliver them Home

Select your Photos and Print them

Who is involved?




Authorization Server (AS)
Obtains Token
Delegates
Uses Token
{ api }
Resource Owner (RO)
Client
Resource Server (RS)

The resource owner

- Has access to some resource or API
- Can delegate access to that resource or API
- Usually has access to a web browser
- Usually is a person

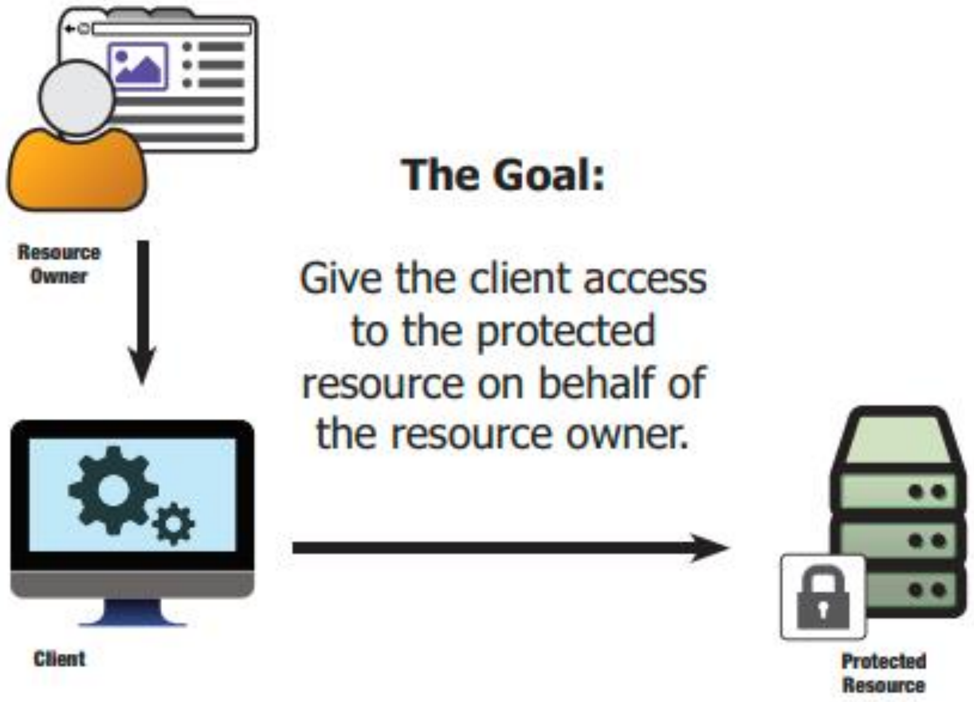
The protected resource

- Web service (API) with security controls
- Protects things for the resource owner
- Shares things on the resource owner’s request

The client application

- Wants to access the protected resource
- Does things on the resource owner's behalf
- Could be a web server
  - But it's still a "client" in OAuth parlance
  - Could also be a native app or JS app

What are we trying to solve?

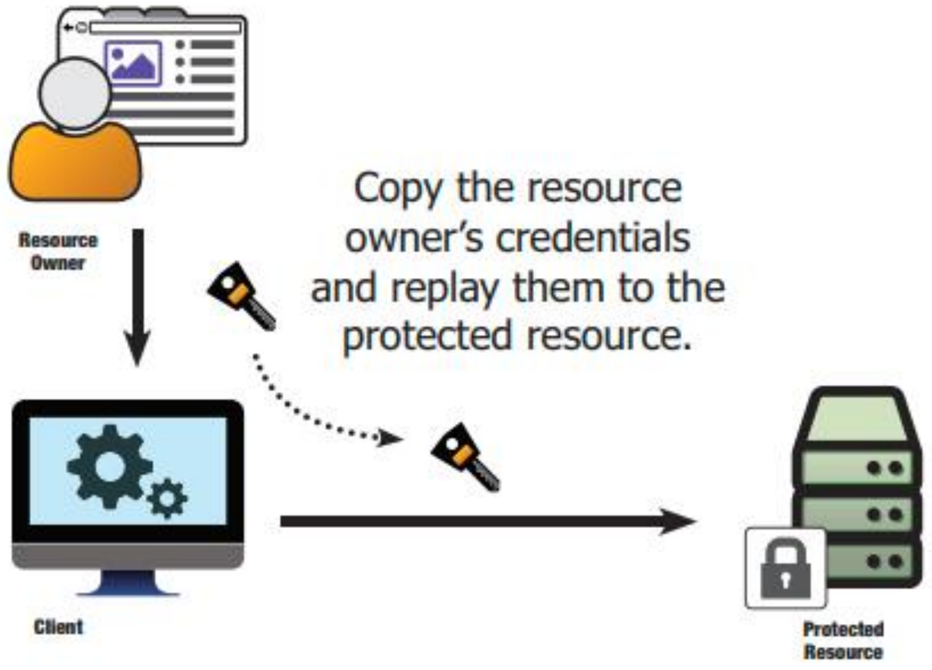


The Goal:

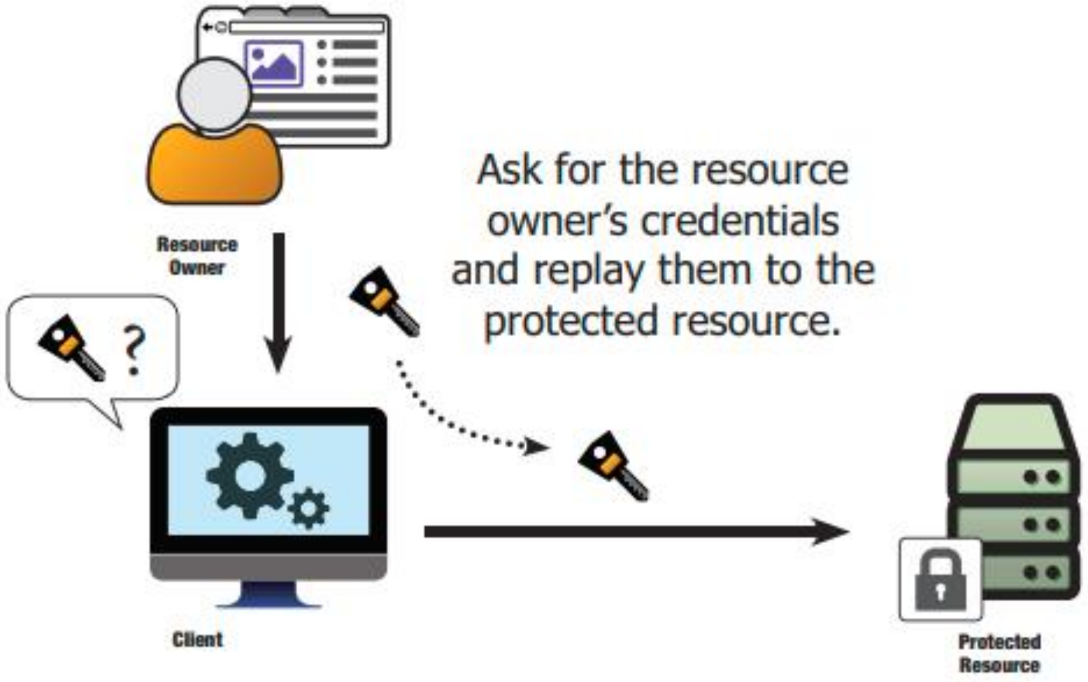
Give the client access to the protected resource on behalf of the resource owner.

- THIS ISN’T A NEW PROBLEM People have been solving this for a long time

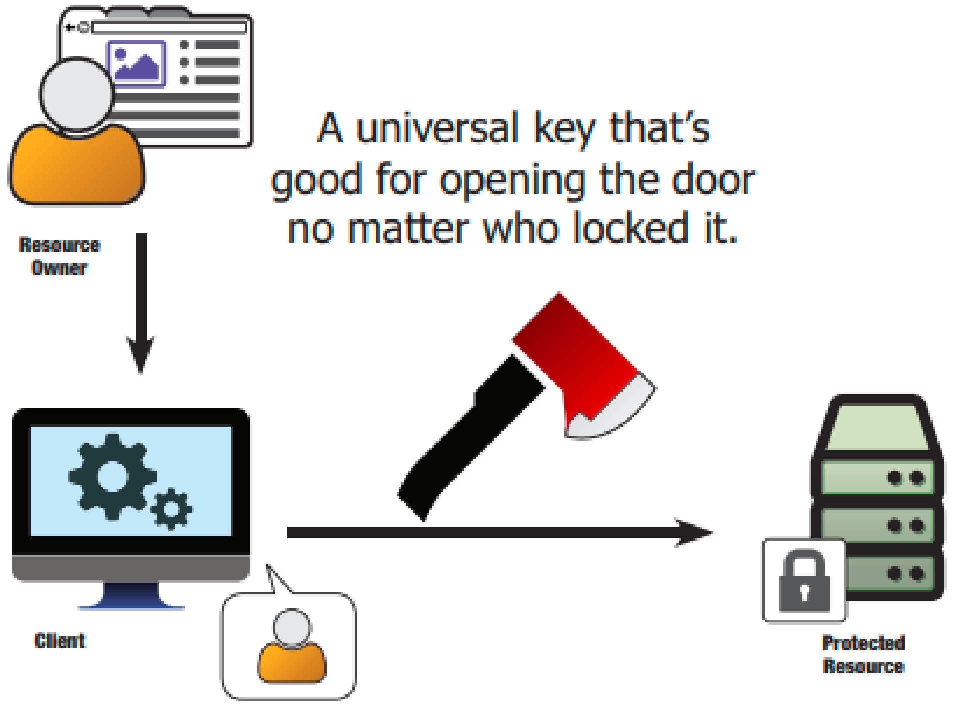
Steal the keys


Resource Owner
Copy the resource owner's credentials and replay them to the protected resource.
Client
Protected Resource

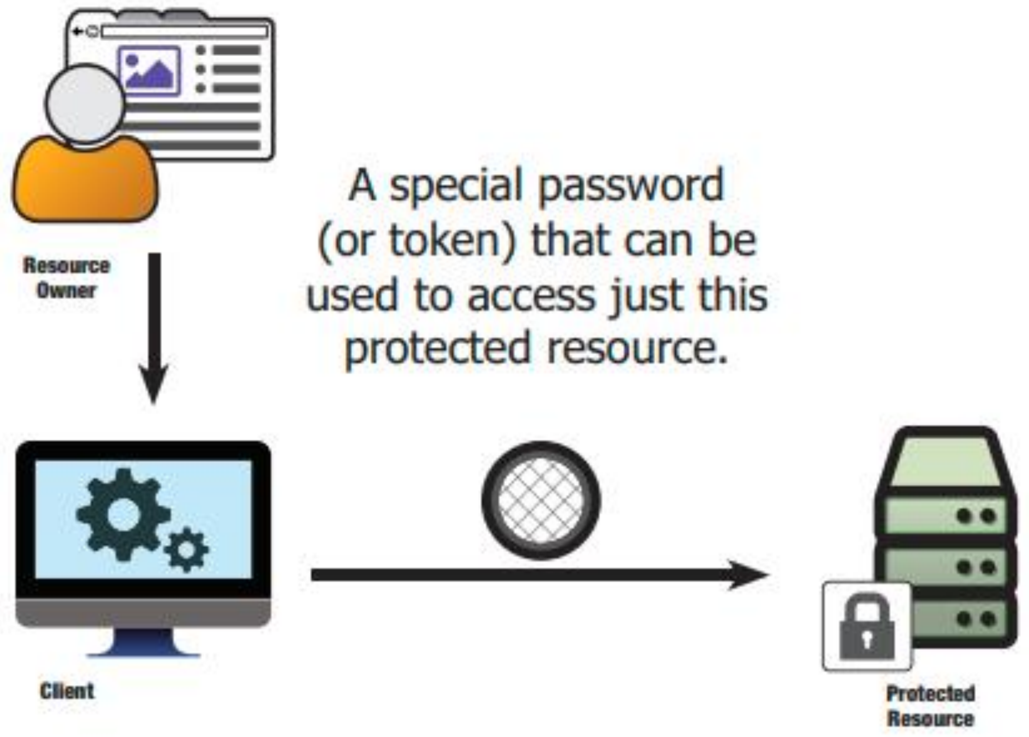
Ask for the keys


Resource Owner
?
Ask for the resource owner's credentials and replay them to the protected resource.
Client
Protected Resource

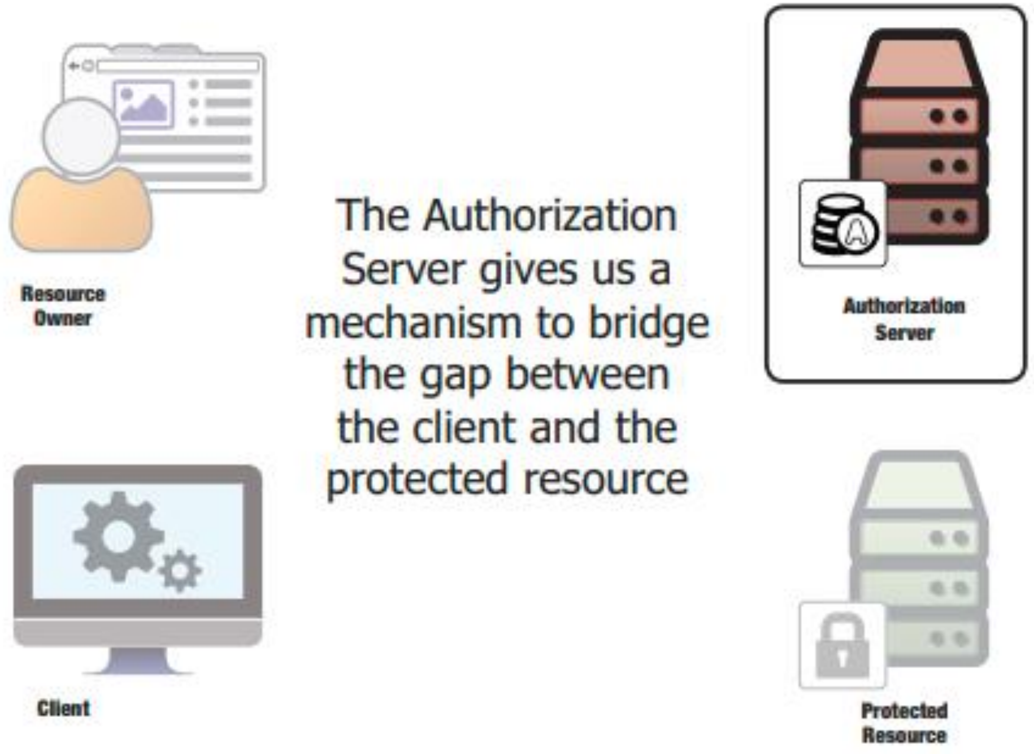
Use a universal key


Resource Owner
A universal key that's good for opening the door no matter who locked it.
Client
Protected Resource

Service-specific credentials


Resource Owner
A special password (or token) that can be used to access just this protected resource.
Client
Protected Resource

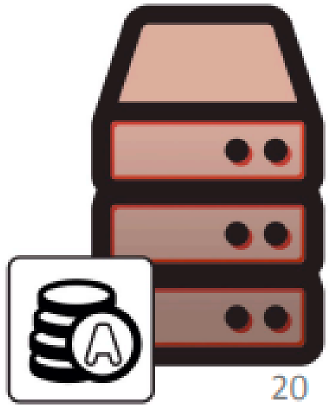
Introducing the Authorization Server (AS)



The Authorization Server gives us a mechanism to bridge the gap between the client and the protected resource

The Authorization Server

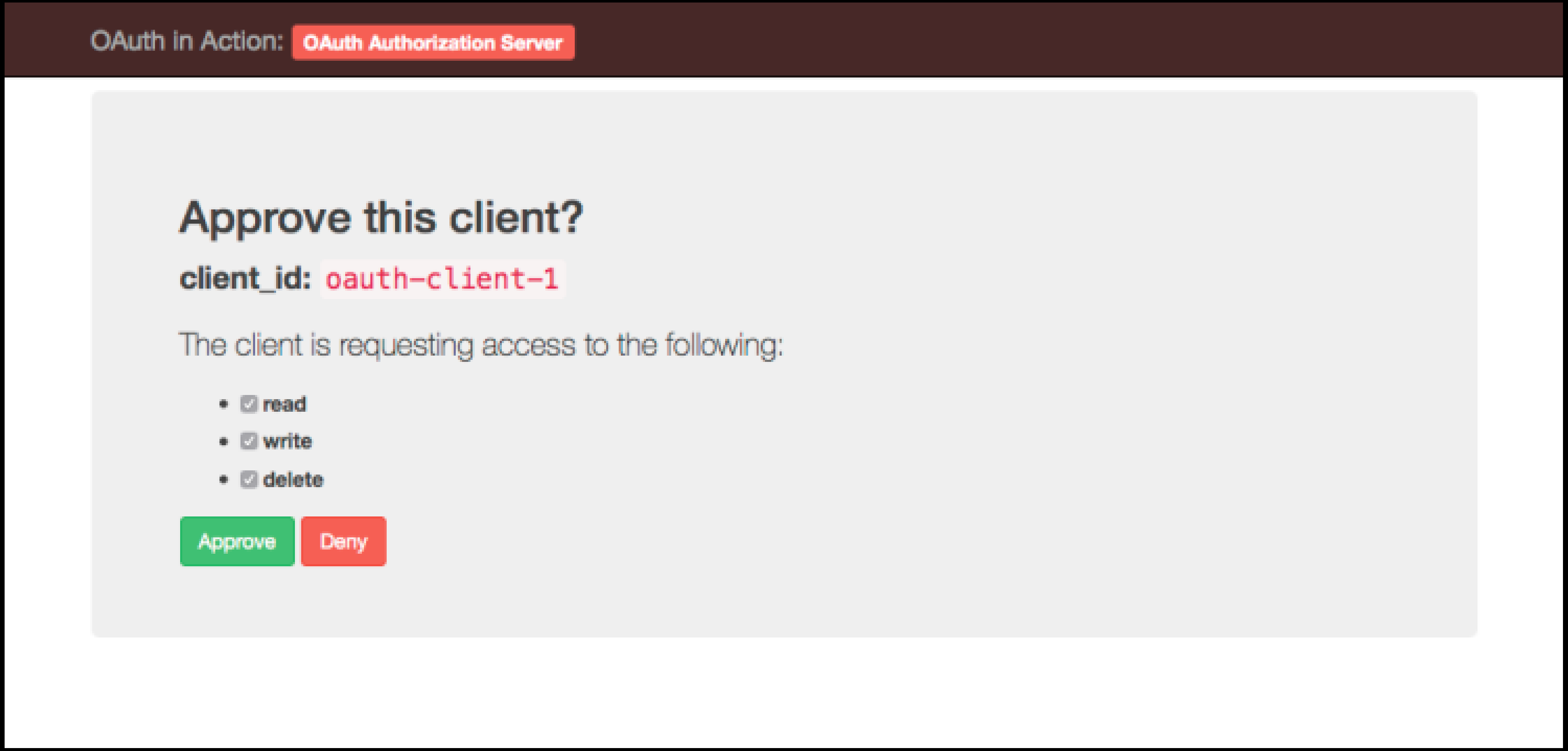
- Generates tokens for the client
- Authenticates resource owners (users)
- Authenticates clients
- Manages authorizations



OAuth Tokens

- Represent granted delegated authorities
  - From the resource owner to the client for the protected resource
- Issued by authorization server
- Used by client
  - Format is opaque to clients
- Consumed by protected resource

You’ve used OAuth



OAuth in Action: OAuth Authorization Server

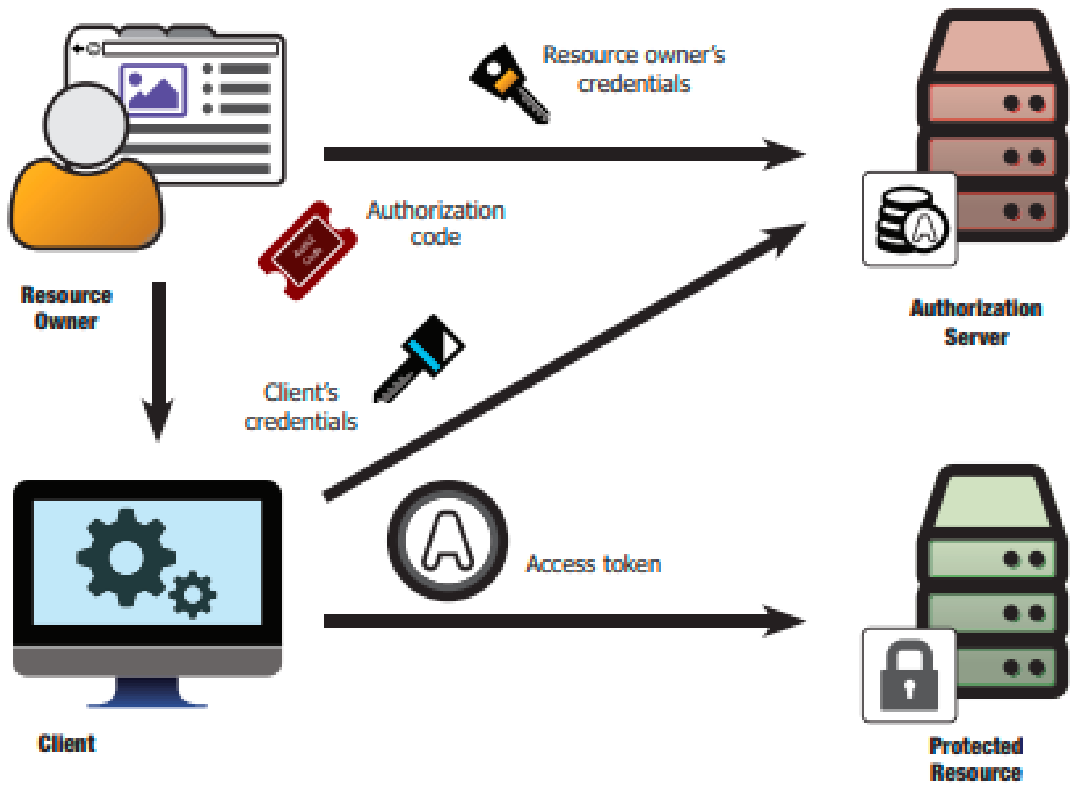
Approve this client?

client_id: `oauth-client-1`

The client is requesting access to the following:

- ✓ read
- ✓ write
- ✓ delete

The authorization code flow



Key Components

- **Resource Owner**: The user who owns the data
- **Client**: The application requesting access
- **Authorization Server**:
- Authenticates the user and issues tokens
- **Resource Server**: Hosts the protected resources

Access Token and Refresh Token

- **Access Token**: Credential used to access resources
- **Refresh Token**: Used to obtain new access tokens

OAuth 2.0 Grant Types

- Authorization Code Grant
- Implicit Grant
- Resource Owner Password Credentials Grant
- Client Credentials Grant
- Refresh Token Grant