Security

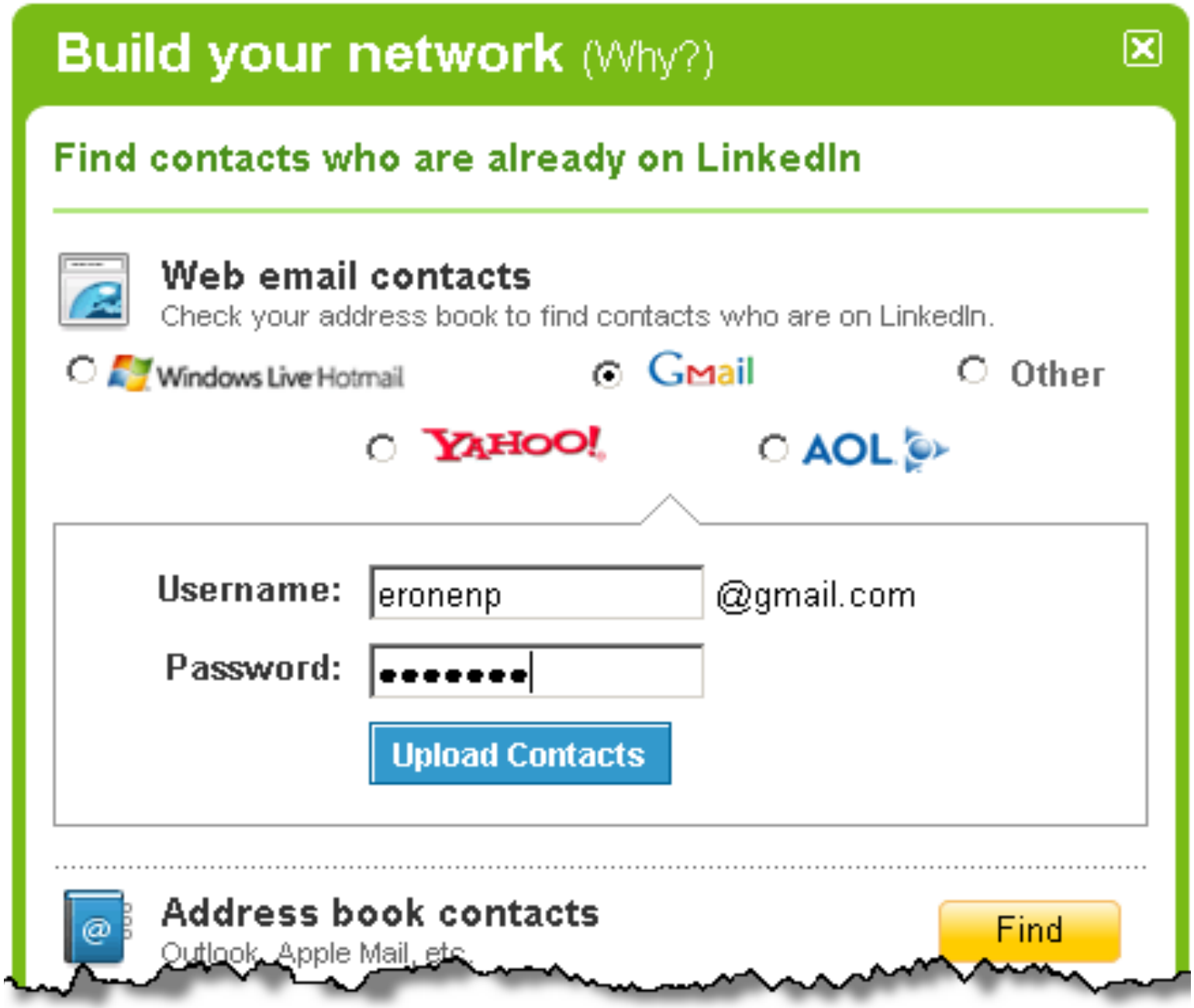
Secure Web Development

– Authorisation and Delegation

Authorisation and delegation: OAuth

OAuth concept

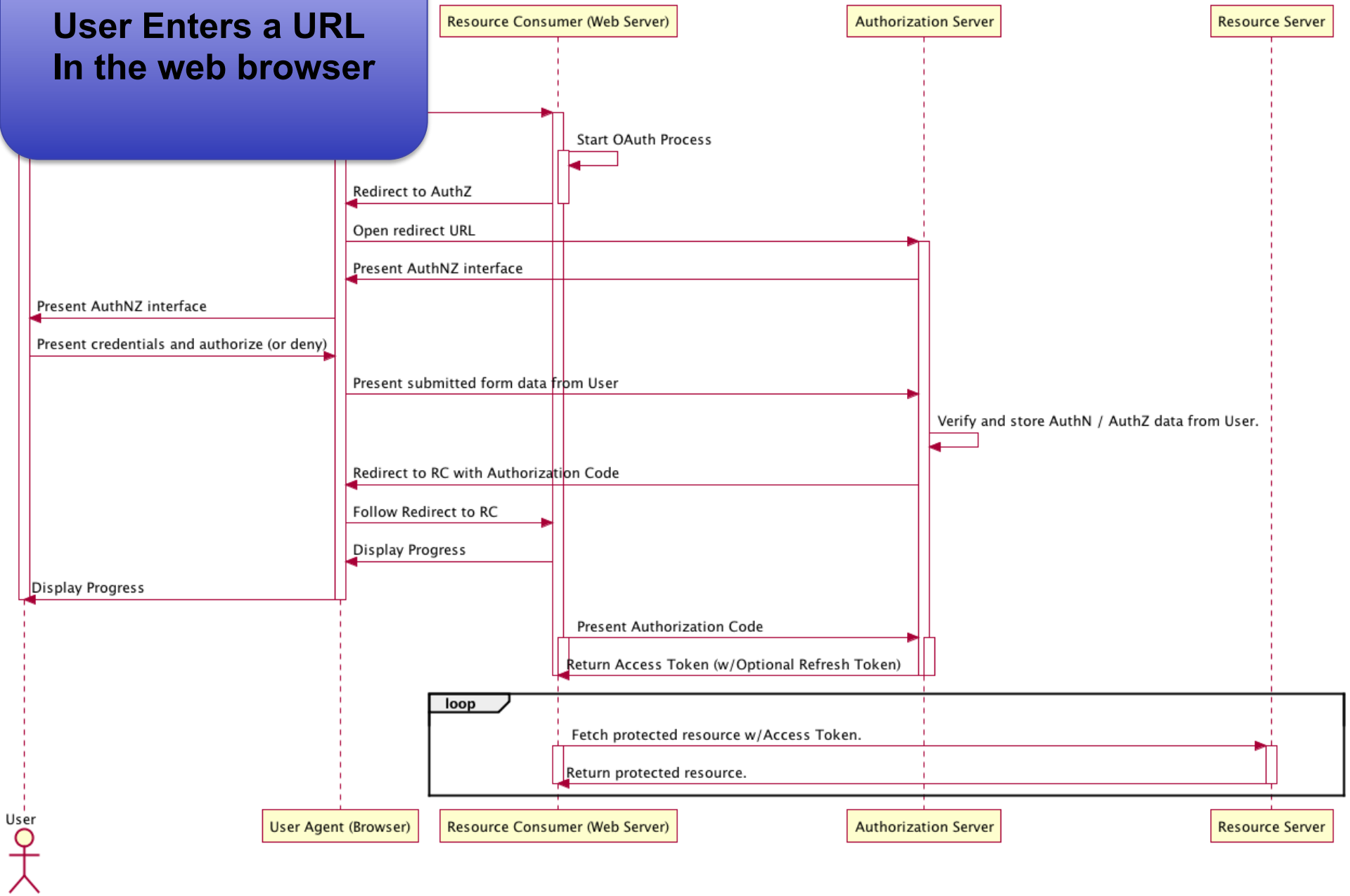
- Open standard for authorisation
- Focused on secure delegation of access
- i.e. I allow a web application to have (perhaps limited) access to another web application
- Based on access tokens
- Related to idea of single sign-on (SSO)

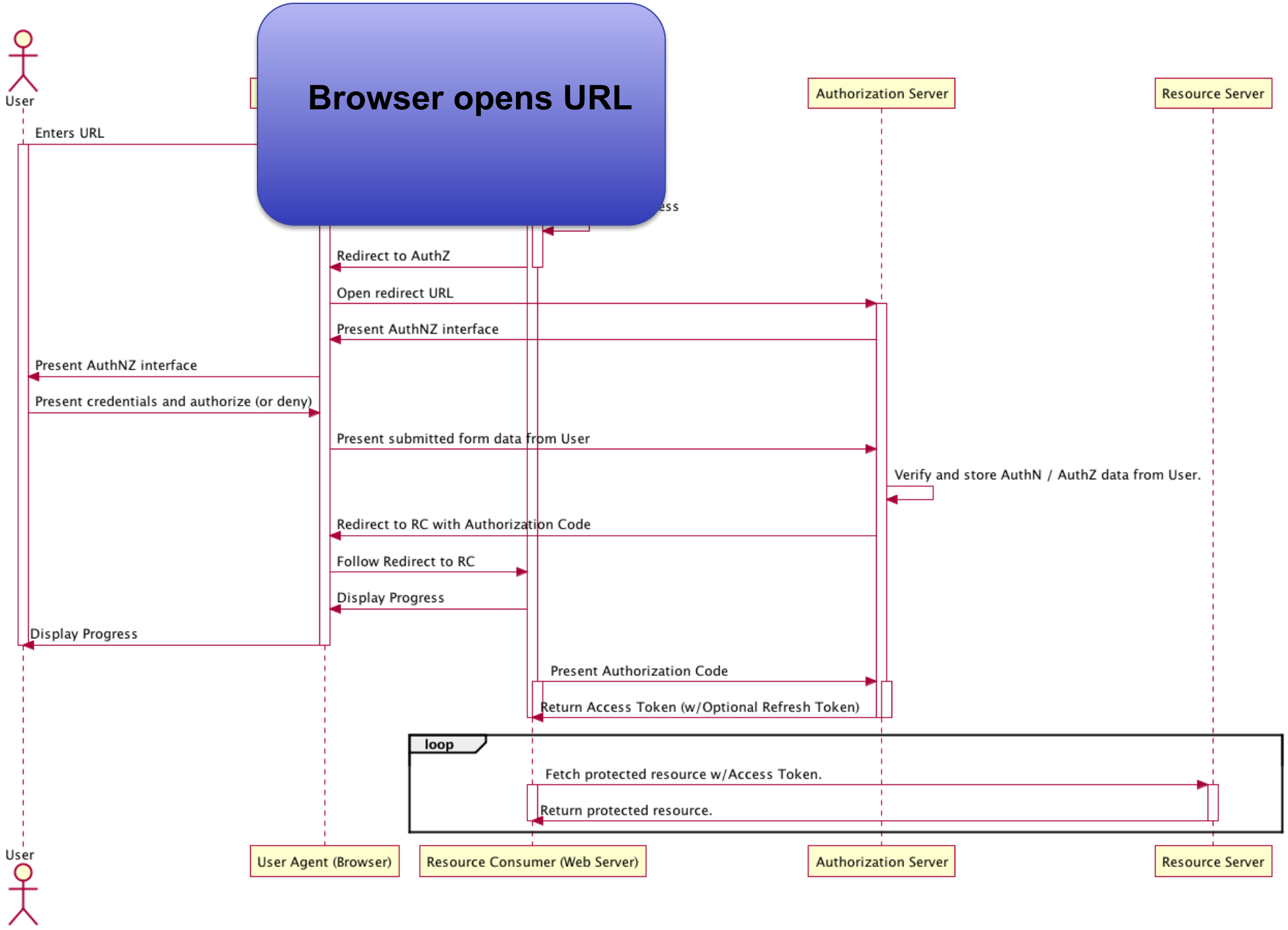

Build your network (Why?)
Find contacts who are already on LinkedIn
Web email contacts
Check your address book to find contacts who are on LinkedIn.
Windows Live Hotmail
Gmail
Other
YAHOO!
AOL
Username:
eronenp
@gmail.com
Password:
Upload Contacts
Address book contacts
Outlook, Apple Mail, etc.
Find

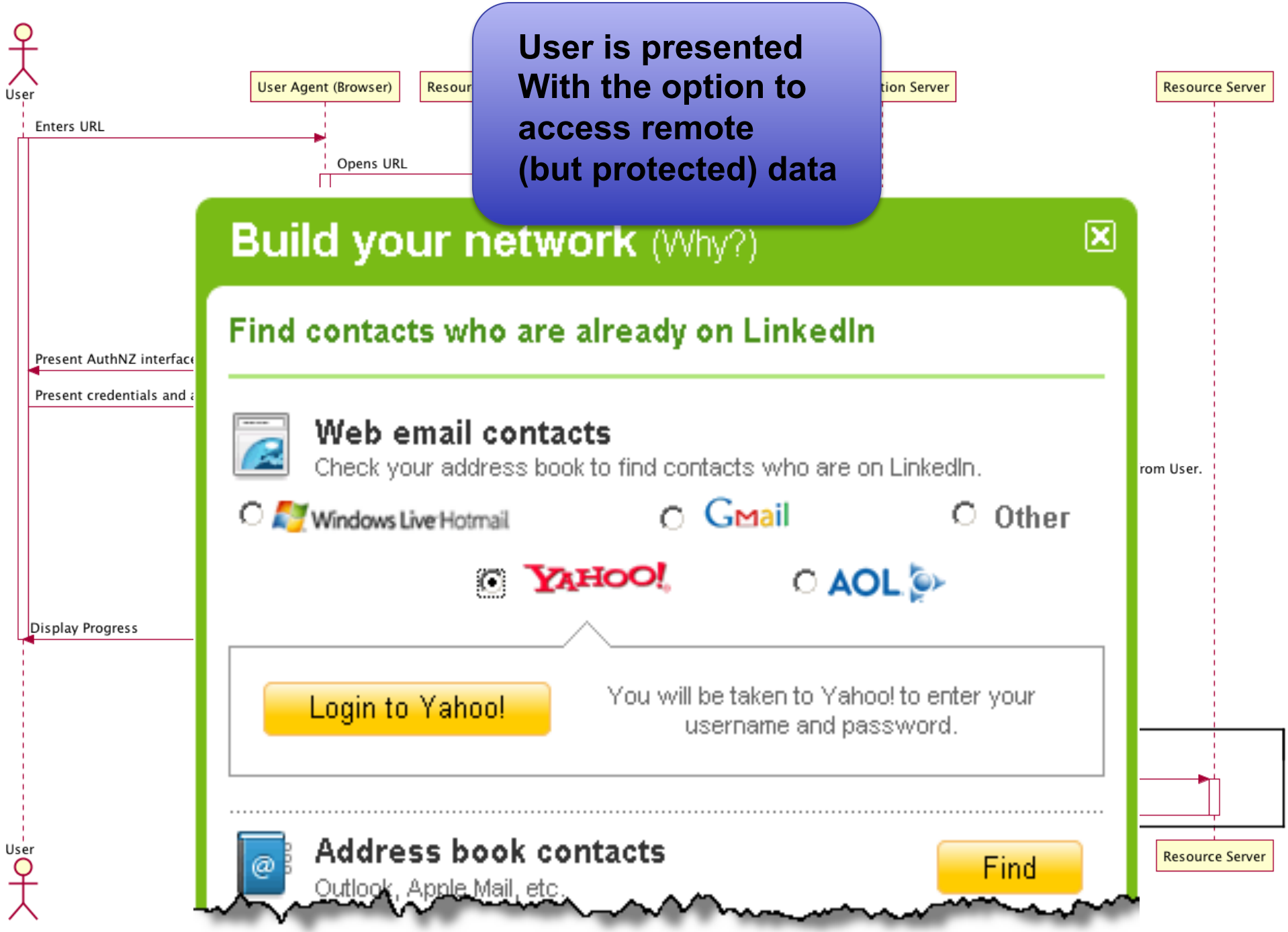
Example OAuth Exchange

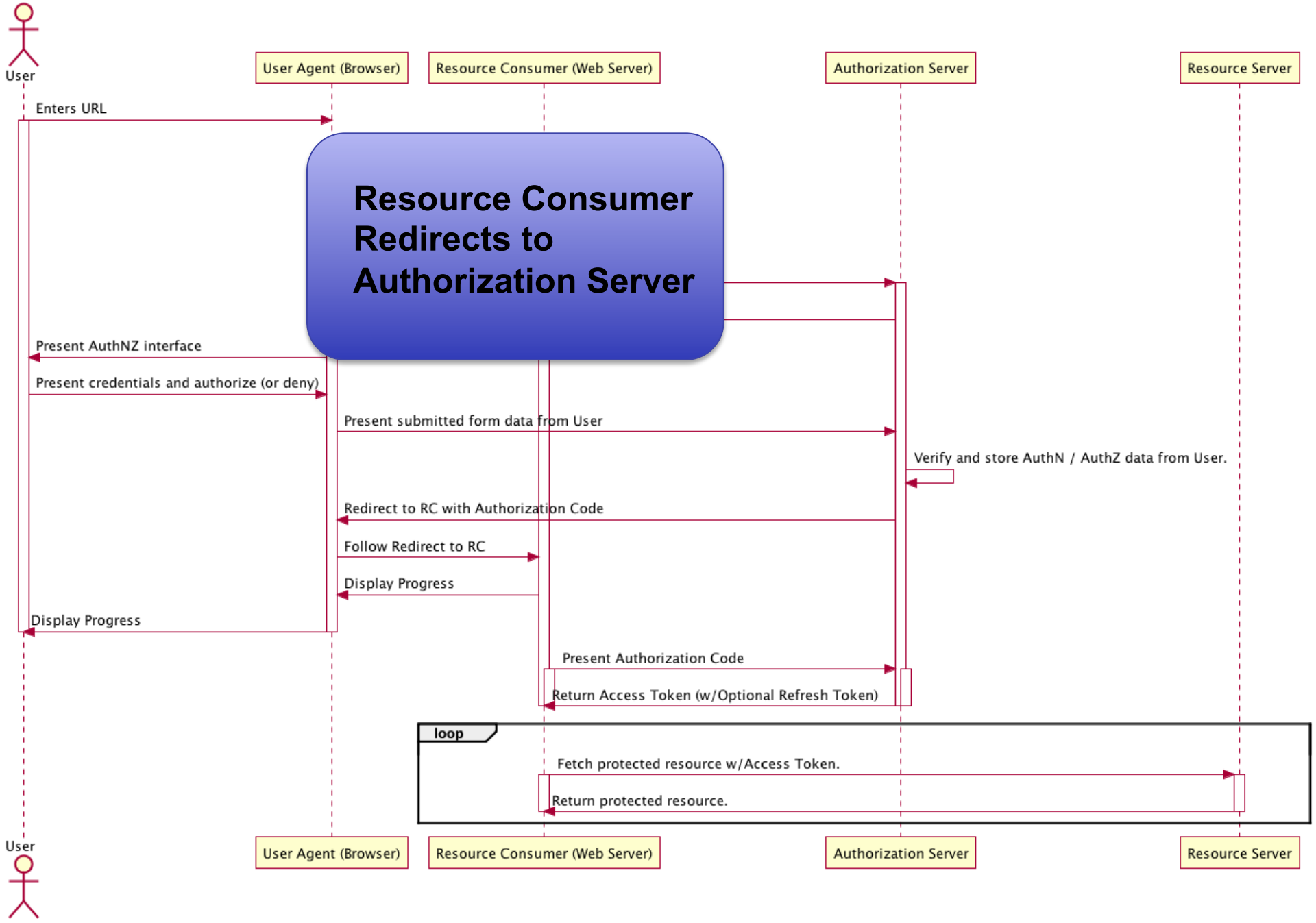
(slides adapted from IETF tutorial by
H Tschofenig & B Cook)

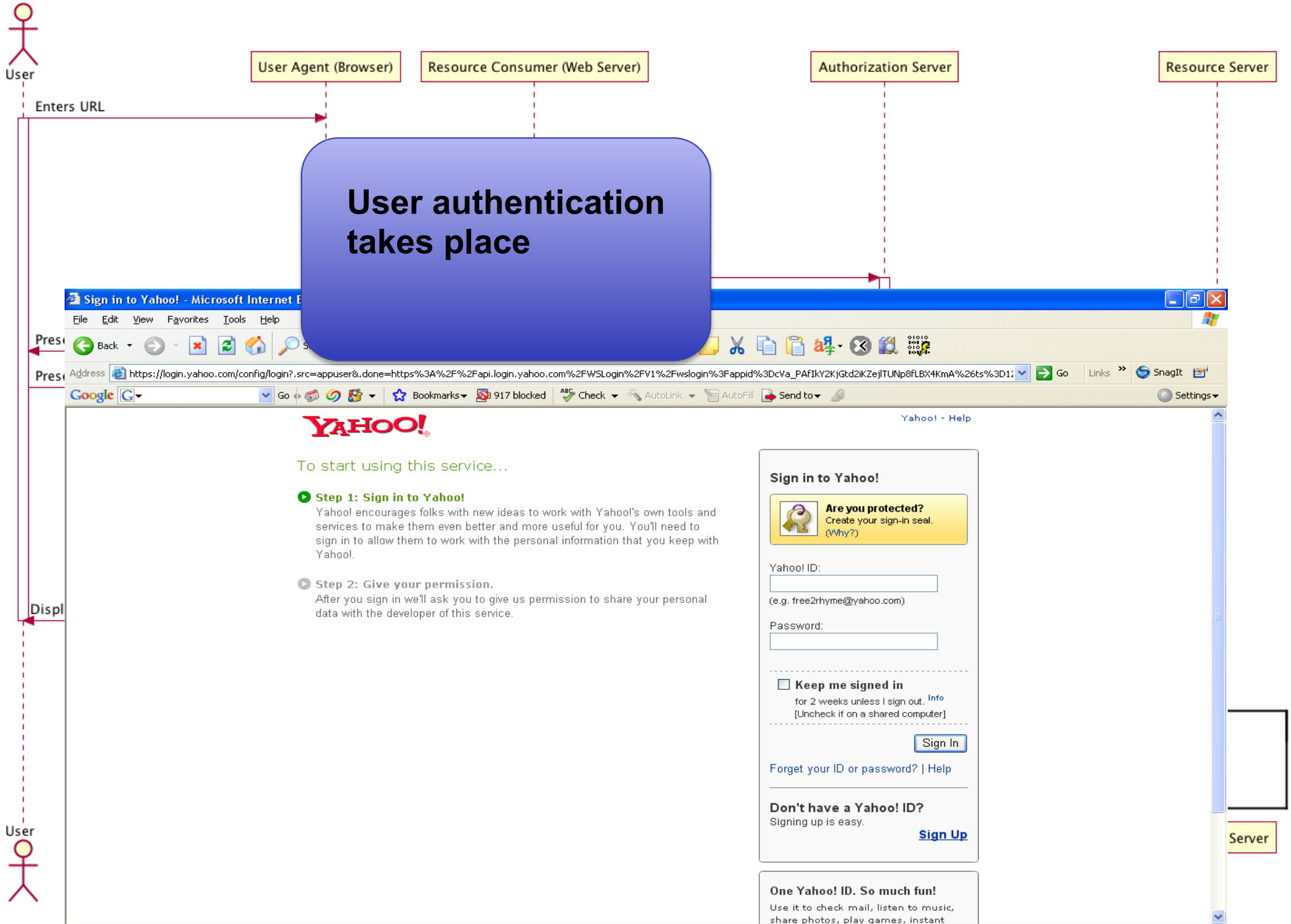
User Enters a URL In the web browser

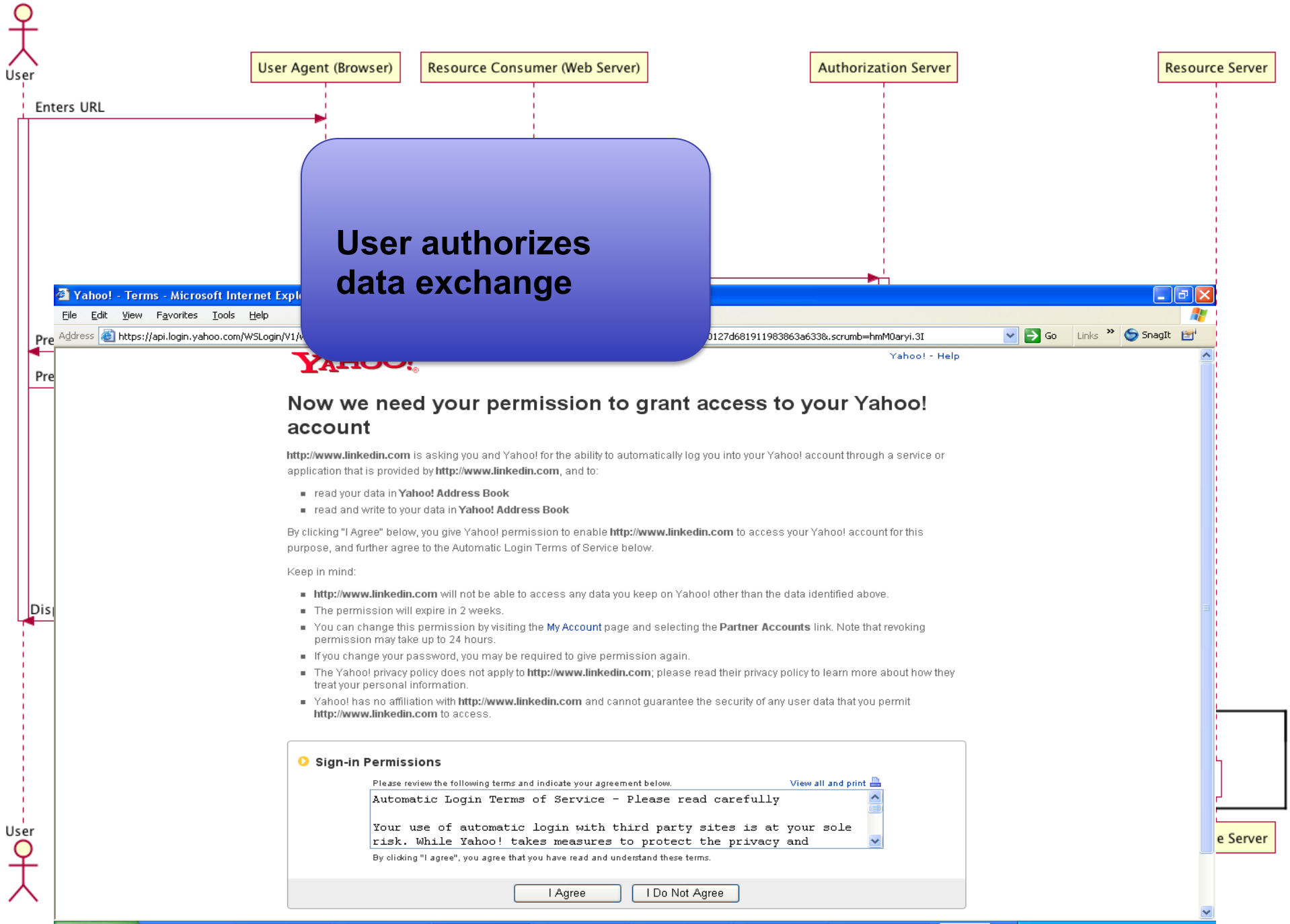


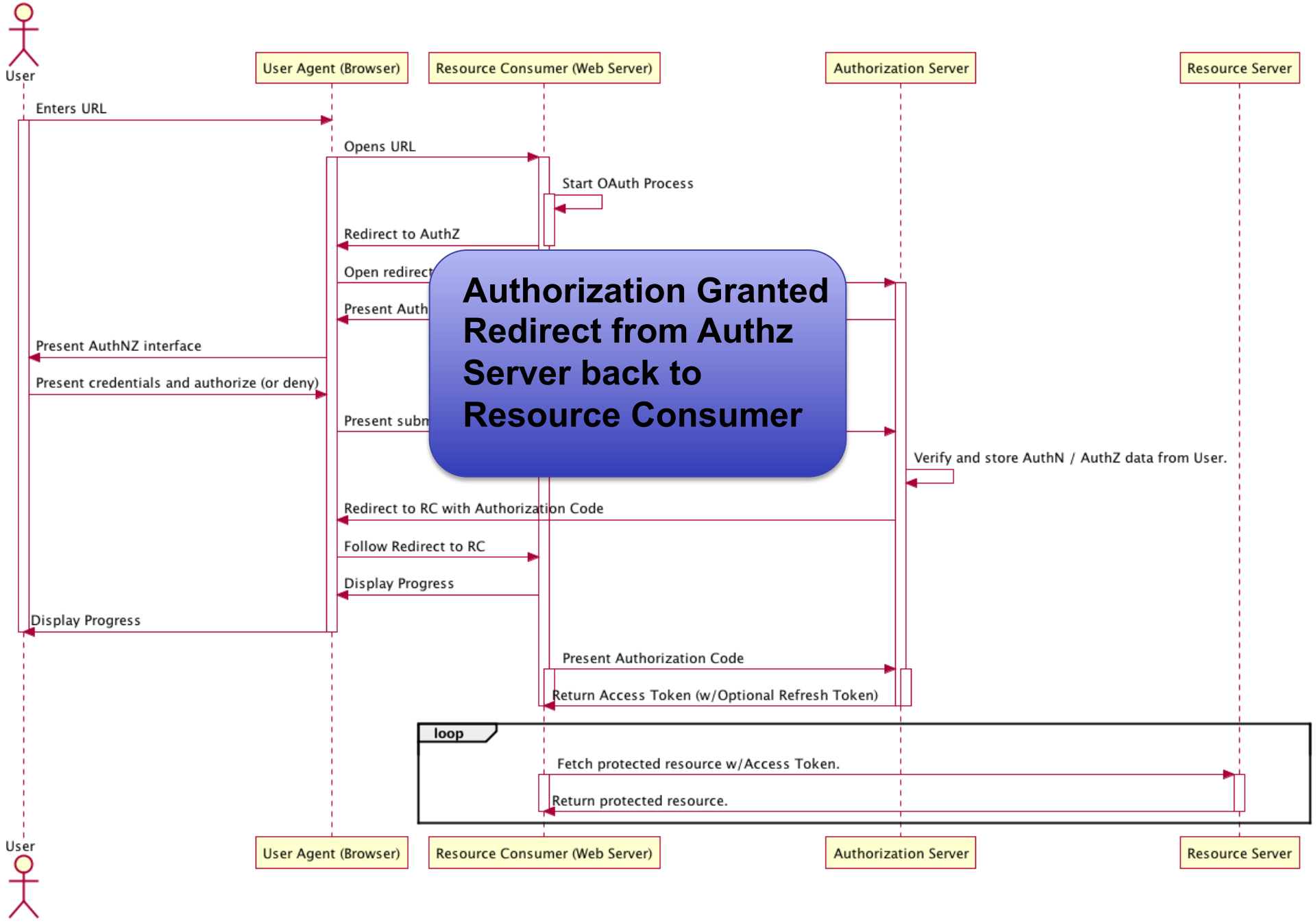

Browser opens URL
User
Authorization Server
Resource Server
Enters URL
Redirect to AuthZ
Open redirect URL
Present AuthNZ interface
Present AuthNZ interface
Present credentials and authorize (or deny)
Present submitted form data from User
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Authorization Code
Follow Redirect to RC
Display Progress
Display Progress
Present Authorization Code
Return Access Token (w/Optional Refresh Token)
loop
Fetch protected resource w/Access Token.
Return protected resource.
User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server

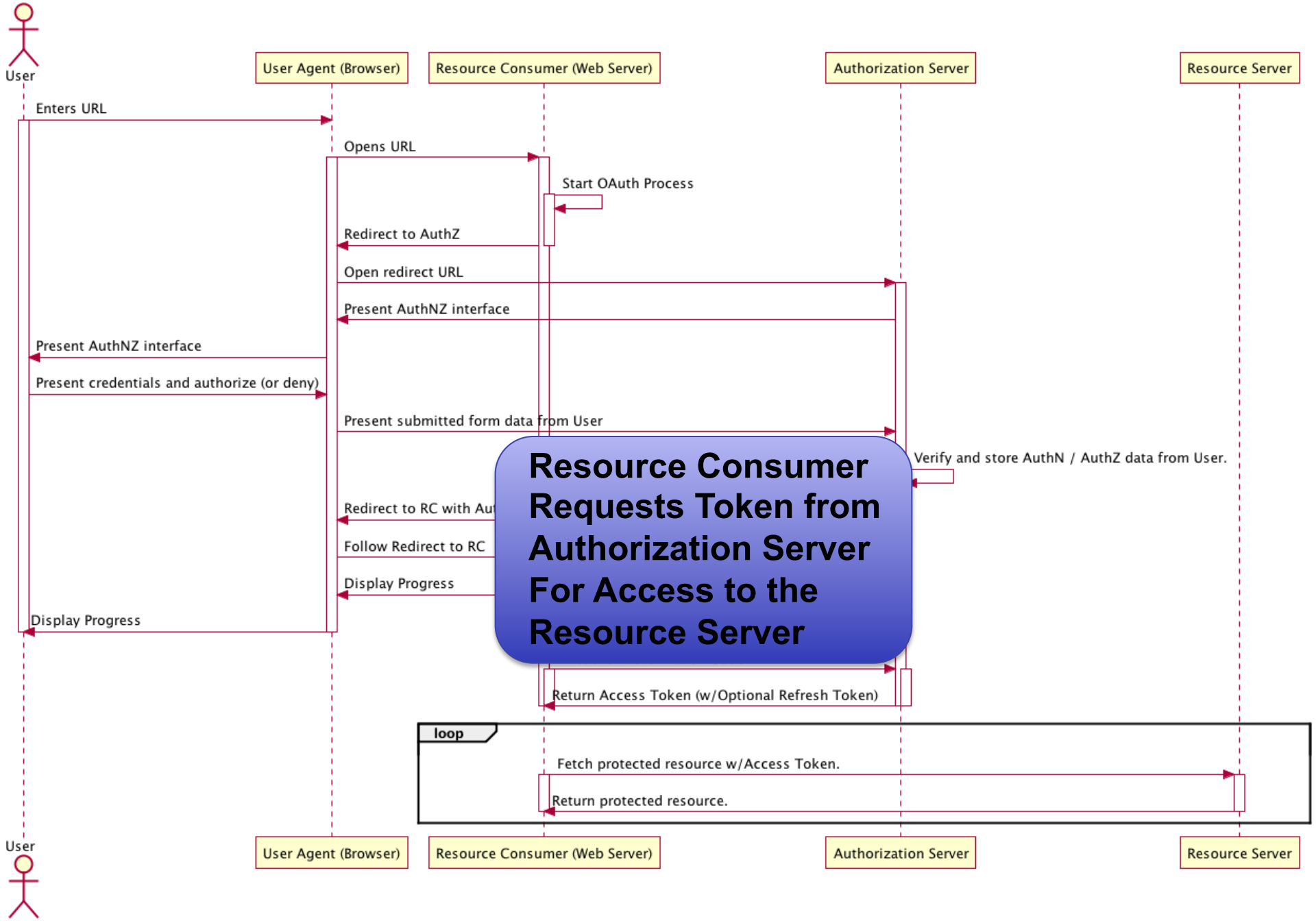

User is presented
With the option to
access remote
(but protected) data
User
User Agent (Browser)
Resour
tion Server
Resource Server
Enters URL
Opens URL
Build your network (Why?)
Find contacts who are already on LinkedIn
Present AuthNZ interface
Present credentials and a
Web email contacts
Check your address book to find contacts who are on LinkedIn.
rom User.
Windows Live Hotmail
Gmail
Other
YAHOO!
AOL
Display Progress
Login to Yahoo!
You will be taken to Yahoo! to enter your username and password.
User
Address book contacts
Outlook, Apple Mail, etc.
Find
Resource Server

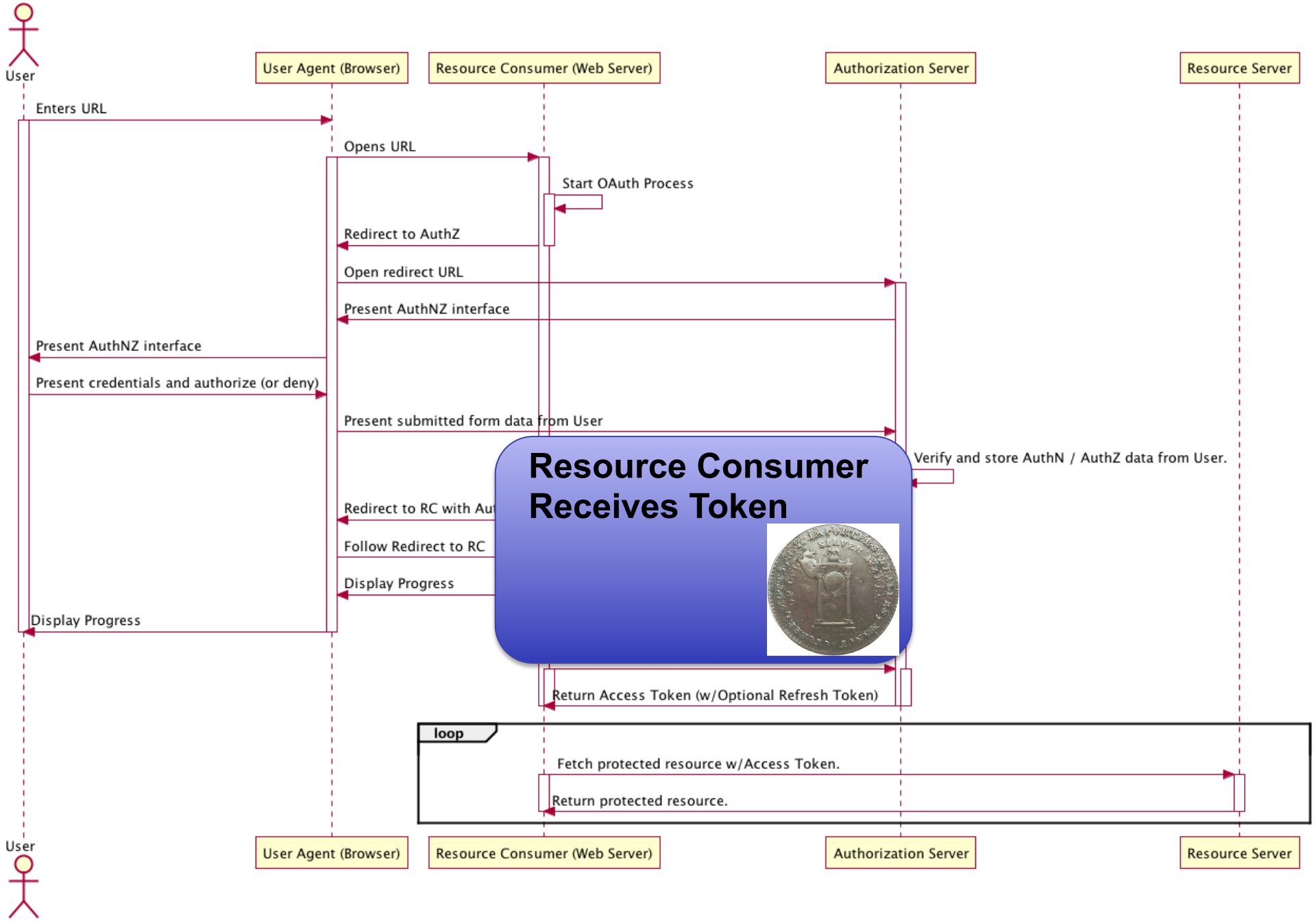

User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
Resource Consumer Redirects to Authorization Server
Present AuthNZ interface
Present credentials and authorize (or deny)
Present submitted form data from User
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Authorization Code
Follow Redirect to RC
Display Progress
Display Progress
Present Authorization Code
Return Access Token (w/Optional Refresh Token)
loop
Fetch protected resource w/Access Token.
Return protected resource.
User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server

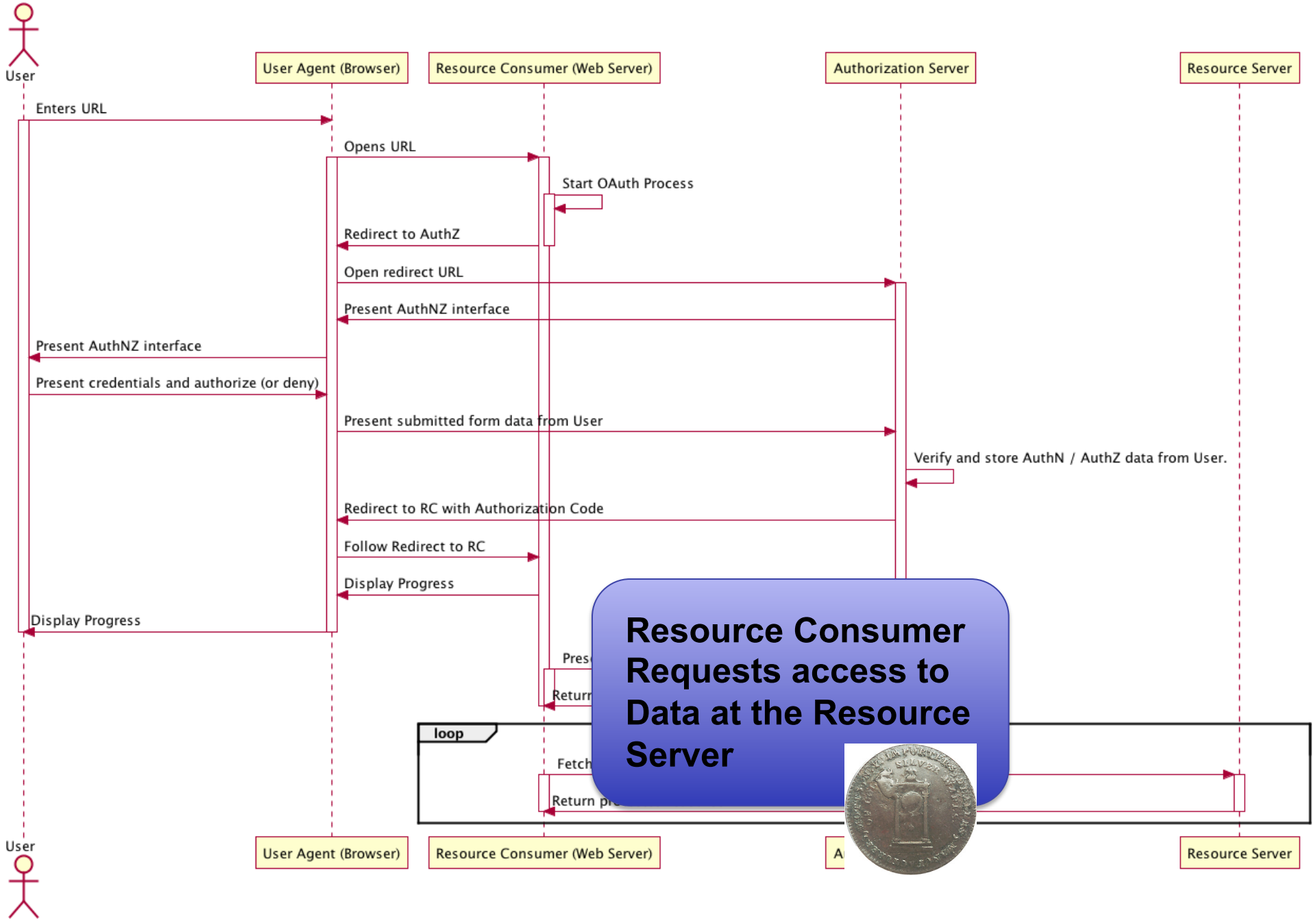

User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
User authentication takes place
Sign in to Yahoo! - Microsoft Internet E
File Edit View Favorites Tools Help
Back
Address https://login.yahoo.com/config/login?.src=appuser&.done=https%3A%2F%2Fapi.login.yahoo.com%2FWSLogin%2FV1%2Fwslogin%3Fappid%3DcVa_PAfIkY2KjGtd2iKZejlTUNp8fLBX4KmA%26ts%3D1
Go
Links
SnagIt
Google
Bookmarks
917 blocked
Check
AutoLink
AutoFill
Send to
Settings
YAHOO!
Yahoo! - Help
To start using this service...
Step 1: Sign in to Yahoo!
Yahoo! encourages folks with new ideas to work with Yahoo!'s own tools and services to make them even better and more useful for you. You'll need to sign in to allow them to work with the personal information that you keep with Yahoo!.
Step 2: Give your permission.
After you sign in we'll ask you to give us permission to share your personal data with the developer of this service.
Sign in to Yahoo!
Are you protected?
Create your sign-in seal.
(Why?)
Yahoo! ID:
(e.g. free2rhyme@yahoo.com)
Password:
Keep me signed in
for 2 weeks unless I sign out. Info
[Uncheck if on a shared computer]
Sign In
Forget your ID or password? | Help
Don't have a Yahoo! ID?
Signing up is easy.
Sign Up
One Yahoo! ID. So much fun!
Use it to check mail, listen to music, share photos, play games, instant
User
Server

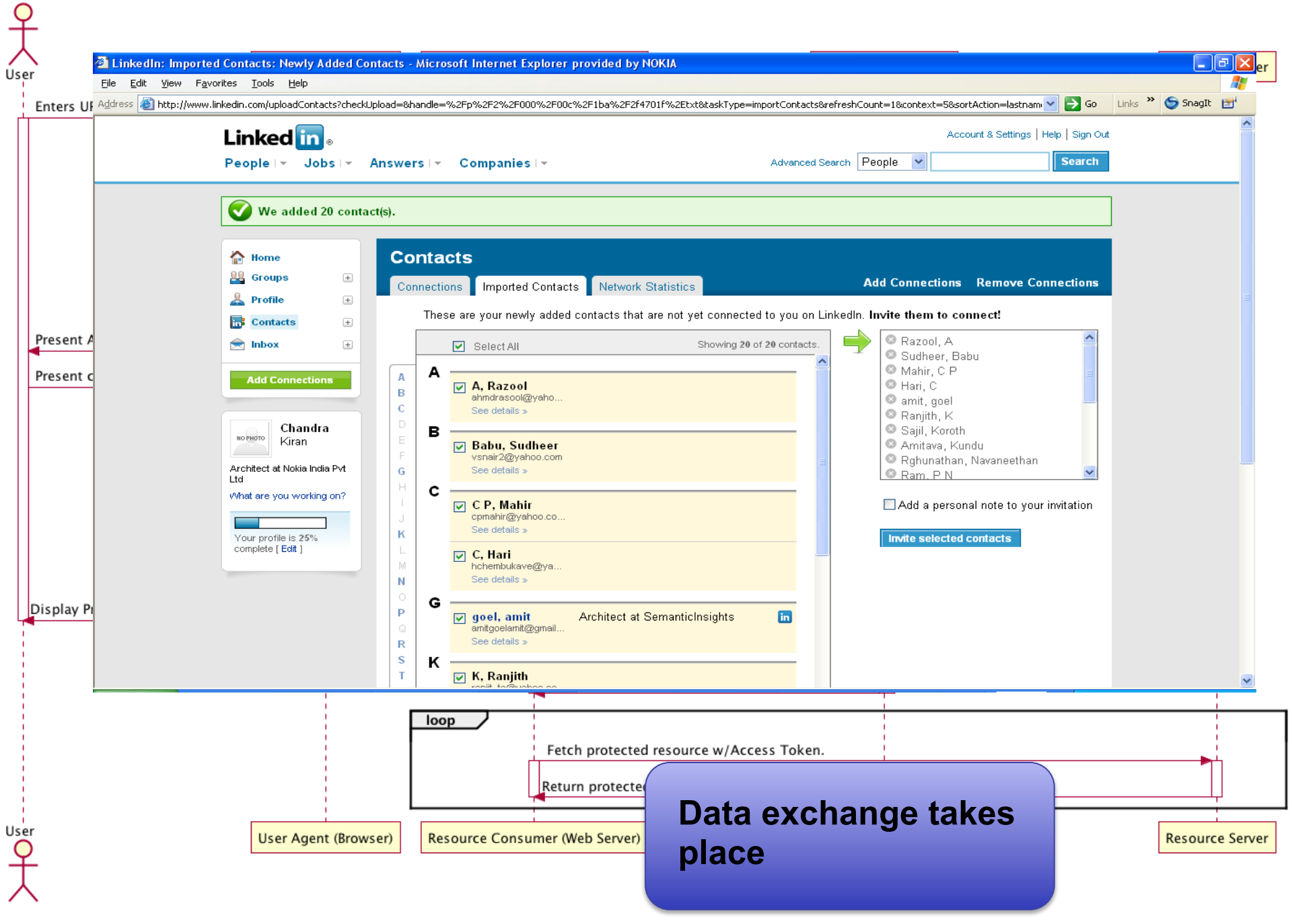

User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
User authorizes data exchange
Yahoo! - Terms - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address https://api.login.yahoo.com/WSLogin/V1/
Go
Links
SnagIt
Yahoo! - Help
Now we need your permission to grant access to your Yahoo! account
http://www.linkedin.com is asking you and Yahoo! for the ability to automatically log you into your Yahoo! account through a service or application that is provided by http://www.linkedin.com, and to:
read your data in Yahoo! Address Book
read and write to your data in Yahoo! Address Book
By clicking "I Agree" below, you give Yahoo! permission to enable http://www.linkedin.com to access your Yahoo! account for this purpose, and further agree to the Automatic Login Terms of Service below.
Keep in mind:
http://www.linkedin.com will not be able to access any data you keep on Yahoo! other than the data identified above.
The permission will expire in 2 weeks.
You can change this permission by visiting the My Account page and selecting the Partner Accounts link. Note that revoking permission may take up to 24 hours.
If you change your password, you may be required to give permission again.
The Yahoo! privacy policy does not apply to http://www.linkedin.com; please read their privacy policy to learn more about how they treat your personal information.
Yahoo! has no affiliation with http://www.linkedin.com and cannot guarantee the security of any user data that you permit http://www.linkedin.com to access.
Sign-in Permissions
Please review the following terms and indicate your agreement below.
View all and print
Automatic Login Terms of Service - Please read carefully
Your use of automatic login with third party sites is at your sole risk. While Yahoo! takes measures to protect the privacy and
By clicking "I agree", you agree that you have read and understand these terms.
I Agree
I Do Not Agree
User
e Server


User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
Opens URL
Start OAuth Process
Redirect to AuthZ
Open redirect
Present Auth
Present AuthNZ interface
Present credentials and authorize (or deny)
Present subm
Authorization Granted
Redirect from Authz
Server back to
Resource Consumer
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Authorization Code
Follow Redirect to RC
Display Progress
Display Progress
Present Authorization Code
Return Access Token (w/Optional Refresh Token)
loop
Fetch protected resource w/Access Token.
Return protected resource.
User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server


User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
Opens URL
Start OAuth Process
Redirect to AuthZ
Open redirect URL
Present AuthNZ interface
Present AuthNZ interface
Present credentials and authorize (or deny)
Present submitted form data from User
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Au
Follow Redirect to RC
Display Progress
Display Progress
Resource Consumer Requests Token from Authorization Server For Access to the Resource Server
Return Access Token (w/Optional Refresh Token)
loop
Fetch protected resource w/Access Token.
Return protected resource.
User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server


User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
Opens URL
Start OAuth Process
Redirect to AuthZ
Open redirect URL
Present AuthNZ interface
Present AuthNZ interface
Present credentials and authorize (or deny)
Present submitted form data from User
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Au
Follow Redirect to RC
Display Progress
Display Progress
Resource Consumer Receives Token
Return Access Token (w/Optional Refresh Token)
loop
Fetch protected resource w/Access Token.
Return protected resource.
User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server


User
User Agent (Browser)
Resource Consumer (Web Server)
Authorization Server
Resource Server
Enters URL
Opens URL
Start OAuth Process
Redirect to AuthZ
Open redirect URL
Present AuthNZ interface
Present AuthNZ interface
Present credentials and authorize (or deny)
Present submitted form data from User
Verify and store AuthN / AuthZ data from User.
Redirect to RC with Authorization Code
Follow Redirect to RC
Display Progress
Display Progress
loop
Resource Consumer Requests access to Data at the Resource Server
User
User Agent (Browser)
Resource Consumer (Web Server)
Resource Server

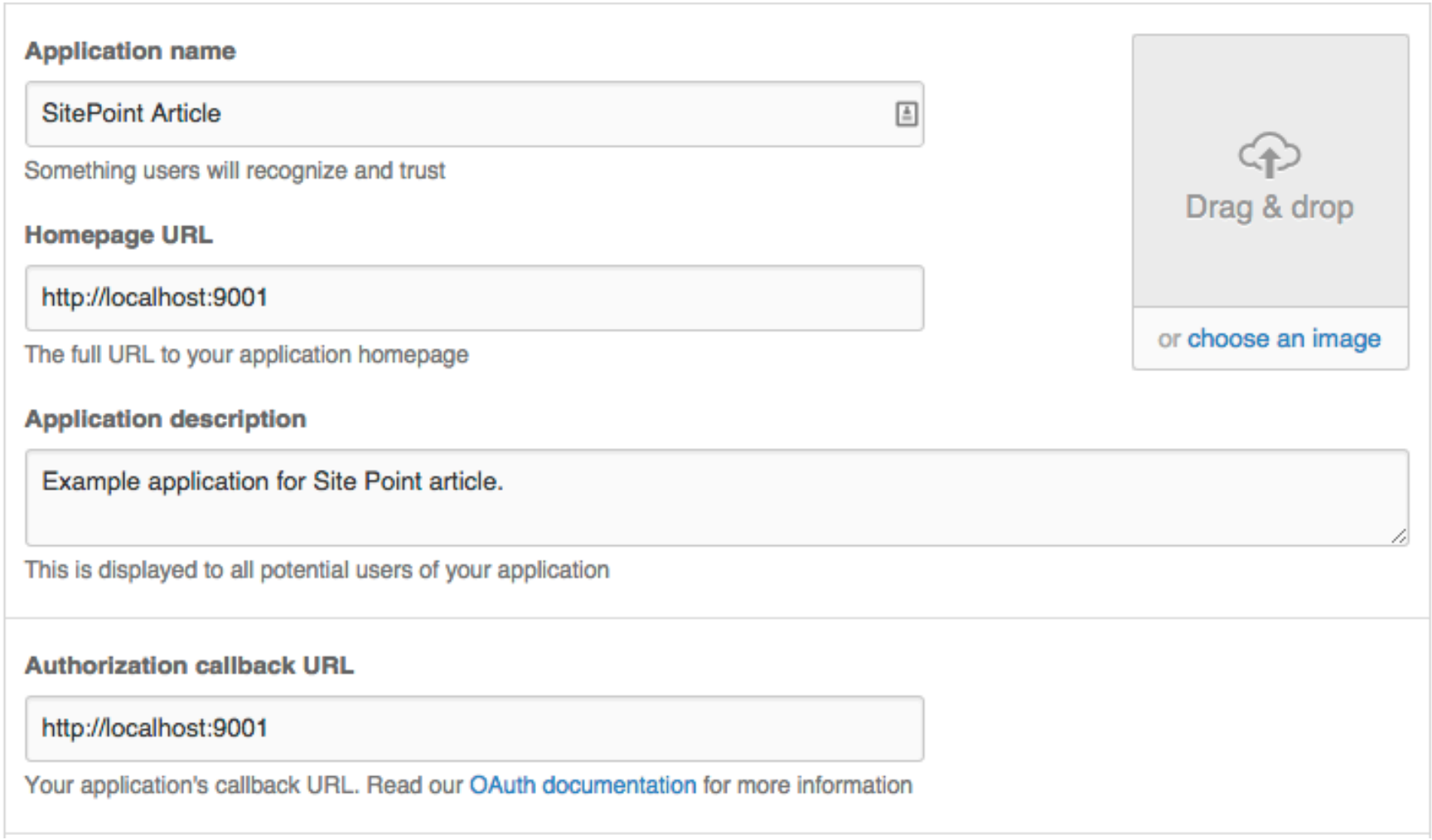

User
Enters UI
Present A
Present c
Display Pr
User
LinkedIn: Imported Contacts: Newly Added Contacts - Microsoft Internet Explorer provided by NOKIA
File Edit View Favorites Tools Help
Address http://www.linkedin.com/uploadContacts?checkUpload=&handle=%2Fp%2F2%2F000%2F00c%2F1ba%2F2f4701f%2Etxt&taskType=importContacts&refreshCount=1&context=5&sortAction=lastnam
Go Links SnagIt
LinkedIn
Account & Settings | Help | Sign Out
People | Jobs | Answers | Companies | Advanced Search People Search
We added 20 contact(s).
Home
Groups
Profile
Contacts
Inbox
Add Connections
Chandra Kiran
Architect at Nokia India Pvt Ltd
What are you working on?
Your profile is 25% complete [Edit]
Contacts
Connections Imported Contacts Network Statistics
Add Connections Remove Connections
These are your newly added contacts that are not yet connected to you on LinkedIn. Invite them to connect!
Select All
Showing 20 of 20 contacts.
A
A, Razool
ahmdrasool@yaho...
See details »
B
Babu, Sudheer
vsnair2@yahoo.com
See details »
C
C P, Mahir
cpmahir@yahoo.co...
See details »
C, Hari
hchembukave@ya...
See details »
G
goel, amit
Architect at SemanticInsights
amitgoelamit@gmail...
See details »
K
K, Ranjith
A B C D E F G H I J K L M N O P Q R S T
Razool, A
Sudheer, Babu
Mahir, C P
Hari, C
amit, goel
Ranjith, K
Sajil, Koroth
Amitava, Kundu
Rghunathan, Navaneethan
Ram, P N
Add a personal note to your invitation
Invite selected contacts
loop
Fetch protected resource w/Access Token.
Return protecte
User Agent (Browser)
Resource Consumer (Web Server)
Resource Server
Data exchange takes place

OAuth with Hapi

- The **bell** module implements OAuth and has built-in support for many providers, including:
  - Facebook, GitHub, Google, Instagram, LinkedIn, Slack, Twitter, Yahoo, Foursquare, Windows Live, BitBucket, Dropbox, Reddit, Tumblr, Salesforce, Pinterest
- Need to register app with provider and get app credentials - e.g. At apps.twitter.com
- Excellent tutorial at this link:

  https://www.sitepoint.com/oauth-integration-using-hapi/

Registering with OAuth provider



OAuth with bell (from SitePoint tutorial)

```
var Hapi = require('hapi');
var Bell = require('bell');
var AuthCookie = require('hapi-auth-cookie');
var server = new Hapi.Server();

server.connection({ port: 4000 });

server.register([Bell, AuthCookie], function (err) {

    if (err) { … }

    var authCookieOptions = {
        password: 'cookie-encryption-password-secure', //Password used for encryption
        cookie: ',my-auth', // Name of cookie to set
        isSecure: false
    };

    server.auth.strategy('my-cookie', 'cookie', authCookieOptions);

    var bellAuthOptions = {
        provider: 'twitter',
        password: 'twitter-encryption-password-secure', //Password used for encryption
        clientId: 'ID HERE',  //'YourAppId',
        clientSecret: 'SECRET HERE',  //'YourAppSecret',
        isSecure: false
    };

    server.auth.strategy('twitter-oauth', 'bell', bellAuthOptions);

    server.auth.default('my-cookie');
```

OAuth with bell (continued)

```
server.route([{
        method: 'GET',
        path: '/login',
        config: {
            auth: 'twitter-oauth',

            handler: function (request, reply) {
                if (request.auth.isAuthenticated) {
                    request.cookieAuth.set(request.auth.credentials);
                    return reply('Hello ' + request.auth.credentials.profile.displayName);
                }
                reply('Not logged in...').code(401);
            }
        }
    }, {
        method: 'GET',
        path: '/account',
        config: {
            handler: function (request, reply) {
                reply(request.auth.credentials.profile);
            }
        }
}, {
        method: 'GET',
        path: '/',
        config: {
            auth: {
                mode: 'optional'
            },
            handler: function (request, reply) {
                if (request.auth.isAuthenticated) {
                    return reply('welcome back ' + request.auth.credentials.profile.displayName);
                }
                reply();
            }
        }
    }, {
```