Cookies, sessions and third-party authentication

Claudia Hauff
cse1500-ewi@tudelft.nl

TU Delft

Microsoft Edge Dev @MSEdgeDev · Jan 14

The best of both worlds: a Microsoft stack-compatible browser based on the **Chromium** open source project. The wait is almost over. Get the new Microsoft **Edge** tomorrow.

197 793

Firefox
@Firefox24x7

Layoffs loom for Mozilla workers as Firefox loses share - Fast Company

Report: Firefox maker Mozilla is laying off 70 people as it searches for revenu...
The Firefox browser is nowhere near as popular by marketshare as it once was.
fastcompany.com

10:29 AM · Jan 17, 2020 · dlvr.it

Learning goals

- **Decide** for a given usage scenario whether cookies or sessions are suitable
- **Explain and implement** cookie usage
- **Explain and implement** session usage
- **Implement** third-party authentication

Introduction to cookies and sessions

Recall: HTTP

- HTTP is **stateless**
- HTTP requests contain **all necessary information**
- Web servers do not need to keep track of the requests received
- Advantage: simple server architecture
- Disadvantage: clients have to resend **the same information** in every request

We do a lot of things on the web requiring a known state

- bol.com keeps your shopping cart full, even when you leave the website
- nytimes.com keeps track of the number of articles you have already read this month
- twitter.com keeps you logged in when closing the browser tab.

Not the stateless web is the norm, but the stateful one. Cookies and sessions are today's major* technologies to achieve a stateful web.

* MDN begs to differ: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage

Cookies cannot …

- **Execute** programs
- Access information from a **user's hard drive**
- Generate spam
- Be **read by arbitrary parties**
  - Only the server setting the cookie can access it
  - But: beware of **third-party cookies**

https://www.nytimes.com

ENGLISH ESPAÑOL 中文

SUBSCRIBE NOW LOG IN

The New York Times

Friday, January 17, 2020

Today's Paper

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

am **Your Friday Briefing** Here's what you need to know.

The Daily **Listen to 'The Daily'** What will happen next with impeachment?

In the 'Climate Fwd:' Newsletter One thing we can do: Fix recycling.

S&P 500	+0.84% ↑
Dow	+0.92% ↑
Nasdaq	+1.06% ↑

9°C 11° 4° Delft, Netherlands

For Third Time in History, a President Goes on Trial

Somber Ritual Unfolds in Senate as New Ukraine Evidence Emerges

- Chief Justice John Roberts was sworn in as the presiding officer of President Trump's impeachment trial, and senators vowed to deliver "impartial justice."
- Meanwhile, a trove of records handed over by an associate of Rudy Giuliani offered new details about the pressure campaign in Ukraine.

6h ago 909 comments

House impeachment managers read the charges against President Trump to the Senate, and senators took an oath, vowing to do "impartial justice." Doug Mills/The New York Times

Senators May Judge Based on an Incomplete Story

New revelations are still emerging on the facts of the accusations against Mr. Trump.

6h ago

Lawmakers Could Pull the C-Span Plug at Critical Moments

There will be bitter debates and intense legal wrangling. Much of it is likely to happen behind closed doors.

9h ago

Trump Administration Broke Law in Blocking Ukraine Aid, Watchdog Says

The agency said blocking the military assistance violated a law that limits a president's power to withhold money allocated by Congress.

The few times President Trump did anything in public on Thursday, it was clear impeachment was on his mind.

7h ago

Opinion ›

David Brooks

The Bernie Sanders Fallacy

No, Virginia, there is no class war.

9h ago 504 comments

The Editorial Board

What Grade Does Trump's China Trade Deal Deserve? Incomplete

The cost of President Trump's China strategy continues to grow. He has yet to sh… are wort…

7h ago

Jerry Ka… **I'm a Lib… Thinks I… Must Be**

Paul Kru… **Why Do… America… Children**

Kat Jerci… **Director… Characte…**

Shamira Ibrahim

iframe inspector

- ✓ https://www.nytimes.com/
- https://pay.google.com/gp/p/ui/payframe?origin=https://www.nytimes.co…
- https://news.google.com/swg/_/ui/v1/serviceiframe?_=438681
- https://a3013110282.cdn.optimizely.com/client_storage/a3013110282.html
- about:blank
- https://static01.nyt.com/ads/tpc-check.html
- https://5290727.fls.doubleclick.net/activityi;src=5290727;type=allpa0;cat…
- https://adservice.google.com/ddm/fls/i/src=5290727;type=allpa0;cat=nyt…
- https://adservice.google.nl/ddm/fls/i/src=5290727;type=allpa0;cat=nyti-0…
- https://stags.bluekai.com/site/50134?ret=html&phint=regid%3D7946939…

Inspector Console Debugger Style Editor Performance Memory Network Storage Accessibility

responsive design mode

- Cache Storage
- Cookies
  - https://pay.google.com
  - https://news.google.com
  - https://a3013110282.cdn.optimizely.com
  - https://static01.nyt.com
  - https://5290727.fls.doubleclick.net
  - https://adservice.google.com
  - https://adservice.google.nl
  - https://stags.bluekai.com
  - https://www.nytimes.com
- Indexed DB
- Local Storage
- Session Storage

Client-side storage options

Filter items

Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSi…
__gads	.nytimes.com	/	Sun, 16 Jan 2022 …	Fri, 17 Jan 2020 0…	ID=03368d7a47c43894:	false	Unset
_cb_ls	www.nytim…	/	Mon, 15 Feb 2021 …	Fri, 17 Jan 2020 0…	1	false	Unset
_cb_svref	www.nytim…	/	Fri, 17 Jan 2020 1…	Fri, 17 Jan 2020 0…	null	false	Unset
_cb	www.nytim…	/	Mon, 15 Feb 2021 …	Fri, 17 Jan 2020 0…	B92aYWCM7lVzCZURxG	false	Unset
_chartbe…	www.nytim…	/	Mon, 15 Feb 2021 …	Fri, 17 Jan 2020 0…	.1510822365866.1579254723104.0000101010011101.DFigvMDO0CbgBzZgzhBeBcqED_DDmM.6	false	Unset
_chartbe…	www.nytim…	/	Fri, 17 Jan 2020 0…	Fri, 17 Jan 2020 0…	t=BtWStWDlLWFyBo_klxC_bzBaCToiDR&E=31&x=26&c=0.75&y=9124&w=955	false	Unset
_gat_UA…	.nytimes.com	/	Fri, 17 Jan 2020 0…	Fri, 17 Jan 2020 0…	1	false	Unset
_gcl_au	.nytimes.com	/	Thu, 16 Apr 2020 …	Fri, 17 Jan 2020 0…	1.1.252285396.1579254265	false	Unset
b2b_cig_…	.nytimes.com	/	Thu, 16 Apr 2020 …	Fri, 17 Jan 2020 0…	%7B%22isCorpUser%22%3Afalse%7D	false	Unset
edu_cig_…	.nytimes.com	/	Thu, 16 Apr 2020 …	Fri, 17 Jan 2020 0…	%7B%22isEduUser%22%3Atrue%7D	false	Unset
iter_id	.nytimes.com	/	Mon, 14 Jan 2030 …	Fri, 17 Jan 2020 0…	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhaWQiOiI1ZTlxODFmYWVhYzE4ODAwMDFlMTRkYTkiL…	false	Unset
nyt-a	.nytimes.com	/	Sat, 16 Jan 2021 0…	Fri, 17 Jan 2020 0…	qRWalt5ghzBxnyhO0EmP71Gy	false	Unset
nyt-gdpr	.nytimes.com	/	Fri, 17 Jan 2020 1…	Fri, 17 Jan 2020 0…	1	false	Unset
nyt-geo	.nytimes.com	/	Fri, 17 Jan 2020 1…	Fri, 17 Jan 2020 0…	NL	false	Unset
nyt-jkidd	.nytimes.com	/	Sat, 16 Jan 2021 0…	Fri, 17 Jan 2020 0…	uid=79469397&lastRequest=1579254720556&activeDays=%5B0%2C0%2C0%2C0%2C0%2C0%2C0…	false	Unset
nyt-m	.nytimes.com	/	Fri, 17 Jan 2025 0…	Fri, 17 Jan 2020 0…	A160F2F6F7C910674688CA977B17E62B&uuid=s.4b656272-93e9-471c-a30d-e73de4136e6d&fv=i…	false	Unset
nyt-purr	.nytimes.com	/	Sat, 16 Jan 2021 0…	Fri, 17 Jan 2020 0…	cfh	false	Unset
NYT-T	.nytimes.com	/	Mon, 14 Jan 2030 …	Fri, 17 Jan 2020 0…	ok	false	Unset

https://www.nytimes.com

ENGLISH ESPAÑOL 中文

SUBSCRIBE NOW LOG IN

Friday, January 17, 2020

Today's Paper

World U.S. Politics N.Y. Business Op... Magazine T Magazine Real Estate Video

Your Friday Briefing
Here's what you need to know.

S&P 500 +0.84% ↑
Dow +0.92% ↑
Nasdaq +1.06% ↑

9°C
11° 4°
Delft, Netherlands

incognito mode
private mode

For Third Time in H...

Somber Ritual Unfolds in Senate as New Ukraine Evidence Emerges

- Chief Justice John Roberts was sworn in as the presiding officer of President Trump's impeachment trial, and senators vowed to deliver "impartial justice."
- Meanwhile, a trove of records handed over by an associate of Rudy Giuliani offered new details about the pressure campaign in Ukraine.

6h ago 909 comments

House impeachment managers read the charges against President Trump to the Senate, and senators took an oath, vowing to do "impartial justice." Doug Mills/The New York Times

The Bernie Sanders Fallacy
No, Virginia, there is no class war.

The cost of President Trump's China strategy continues to grow. He has yet to show that the benefits are worth it.

8h ago

Senators May Judge Based on an Incomp...
New revelations are still emerging on t... the accusations against Mr. Trump.

6h ago

...uld Pull the C-Span Plug a...
Momen...
There will b... wrangling. Mu... ...s likely to ha... ...nind closed doors.

9h ago

I'm a Lib... Thinks Imm... Must Be Restr...

Paul Krugman
Why Does America Hate It... Children?

Kat Jercich
...Your Gay ...Be Gay

Shamira Ibrahim

Jessica Stern
Why Did I Let a Convicted War Criminal Practice Energy Healing on Me?

David Leonhardt
What Americans Don't Understand About China's Power

Joan C. Williams
The Misguided Push for the Equal Rights Amendment

Trump Administration Broke Law in Blocking Ukraine Aid, Watchdog Says

The agency said blocking the military assistance violated a law that limits a president's power to withhold money allocated by Congress.

The ...nes President Trum... ...nything in public... ...rsday, it was clea... ...chment was on his...

7h ago

request cookies
response cookies

Inspector Console Debugger Style Editor Performance Memory Network Storage Accessibility

Filter URLs | All HTML CSS JS XHR Fonts Images Media WS Other | Persist Logs | Disable cache | No throttling | HAR

Status	Method	Domain	File	Cause	Type	Transferred	Size
200	GET	et.nytimes.com	/?subject=page&url=https://www.nytimes.com/&asset...	script	js	974 B	107 B
200	POST	a.et.nytimes.com	track	beacon	json	881 B	35 B
200	GET	et.nytimes.com	/?subject=page&url=https://www.nytimes.com/&asset...	script	js	974 B	107 B
200	POST	a.et.nytimes.com	track	beacon	json	881 B	35 B
200	GET	platform.iterateh...	sdk-prod-3a1627e9f9145639d326.js	script	js	cached	0 B
200	GET	platform.iterateh...	style-ac9e39081c324e3f7ee774211f57b680.css	stylesheet	css	cached	135.01 KB
200	GET	adservice.google.nl	src=5290727;type=allpa0;cat=nyti-0;ord=1;num=692...	subdocument	html	1.11 KB	194 B
200	GET	www.google-a...	collect?v=1&_v=j79&a=152690447&t=pageview&_s=1...	img	gif	587 B	35 B
200	GET	stags.bluekai....	50136?limit=1&id=qRWalt5ghzBxnyhO0EmP71Gy&gt...	img	gif	739 B	62 B
200	OPTIONS	iteratehq.com	embed	fetch	plain	877 B	0 B
304	GET	sb.scorecardr...	cs.js	script	js	cached	0 B
200	GET	static.chartbe...	chartbeat.js	script	js	cached	0 B
200	POST	iteratehq.com	embed	fetch	json	766 B	14 B
200	GET	pnytimes.char...	ping?h=nytimes.com&p=nytimes.com/&u=B92aYWCM...	img	gif	246 B	43 B

144 requests | 4.20 MB / 301.75 KB transferred | Finish: 6.06 s | DOMContentLoaded: 468 ms | load: 4.43 s

Headers Cookies Params Response Timings Stack T

Filter cookies

Cookies

Response cookies
- nyt-a:
  - domain: nytimes.com
  - expires: 2021-01-16T09:58:03.000Z
  - path: /
  - value: qRWalt5ghzBxnyhO0EmP71Gy
- sessionActive:
  - domain: et.nytimes.com
  - expires: 2020-01-17T10:28:03.000Z
  - path: /
  - value: true

Request cookies
- _gads: ID=03368d7a47c43894:T=1579254262:S=ALNI_MZSjSQxcKC3SQ75ljratwKGfYtf6w
- _gat_UA-58630905-2: 1
- _gcl_au: 1.1.252285396.1579254265
- b2b_cig_opt: {"isCorpUser":false}

Cookies = the server's short term memory

Cookies and sessions are ways to **introduce state** on top of the stateless HTTP.

Cookie: a short amount of text **(key/value)** **sent by the server** and **stored by the client** for some amount of time.

Minimum client storage requirements (`RFC6265`)

- Store at least **4096 bytes** per cookie
- Store at least **50** cookies per domain
- Store at least **3000** cookies total.

They have been around since the early 1990s.

Cookie security

- **Visible** to clients
- Clients can **delete/disallow** them
- Clients can **alter** them
  - Line of attack: **servers** should not send sensitive information in cookies
- **Sessions** are preferable to cookies
  - Sessions make use of cookies
  - Cookie contains only the session identifier

A word of warning: RFC6265

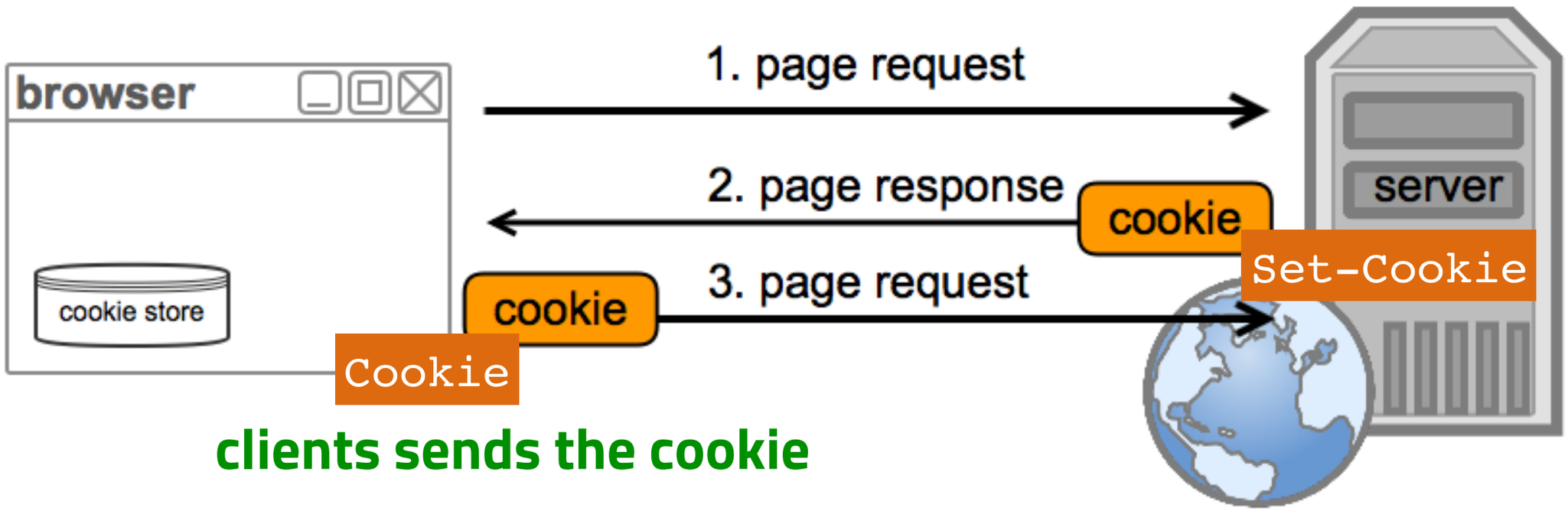
"This document defines the HTTP **`Cookie`** and **`Set-Cookie`** header fields. These header fields can be used by HTTP servers **to store state** (called cookies) at HTTP user agents, letting the servers maintain a **stateful session** over the mostly stateless HTTP protocol.

Although cookies have many historical infelicities that degrade their security and privacy, the `Cookie` and `Set-Cookie` header fields are widely used on the Internet. "

Cookie flow

server sends a cookie once;

resends when key/value changes



clients sends the cookie

back in every request

- Encoded in the **HTTP header**
- Web frameworks offer designated methods
- Cookies are **bound** to a **site domain name** (security feature)

Cookies in more detail

and a bit about Firefox Extensions ....

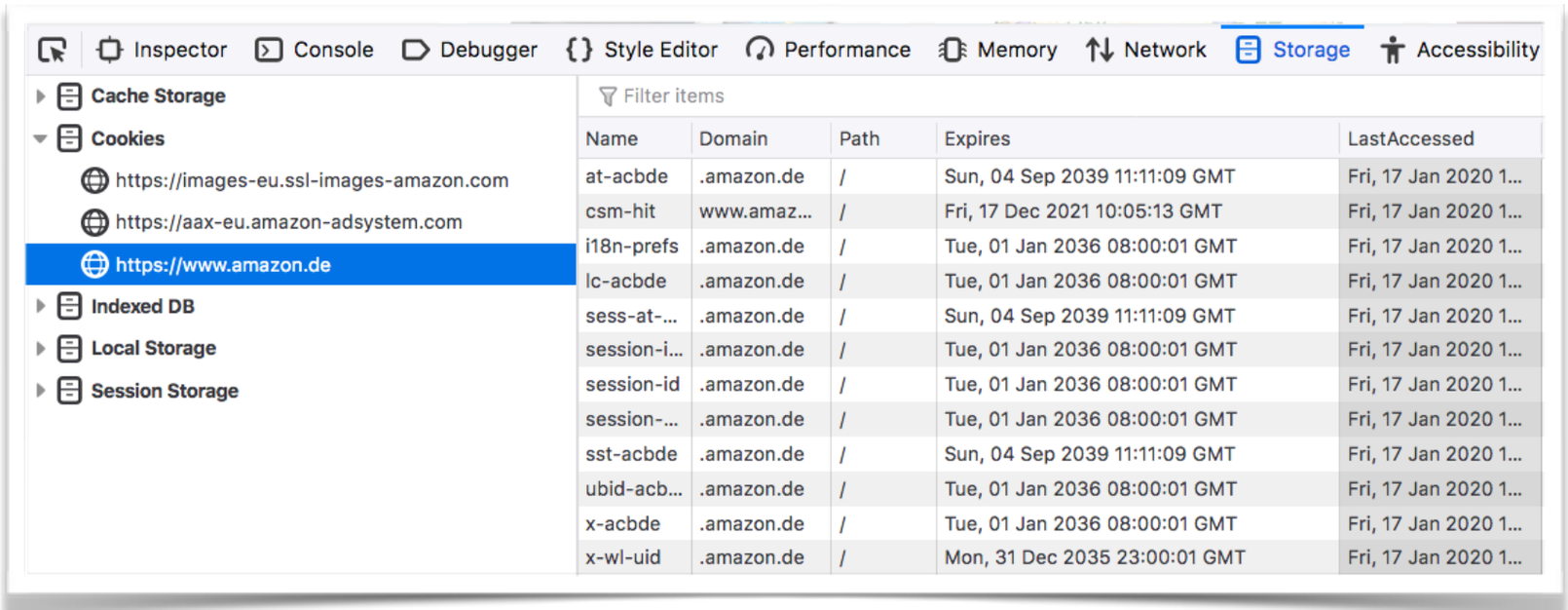
Transient vs. persistent cookies

- **Transient** (or session) cookies:
  - Exist in memory only; deleted when the browser is closed
  - Cookies are transient if **no expiration date** is defined
- **Persistent** cookies:
  - Remain intact after the browser is closed
  - Have a **maximum age**
  - Are **send back to the server** as long as they are **valid**

Not the tab or window!

Cookie fields

- **`cookie-name=cookie-value`** the only **required** field
- **`Expires`** (expiration date) or **`Max-Age`** (*seconds* until it expires)
- **`Domain`** the cookie is associated with; cookies can only be assigned to the **same domain** the server is running on
- **`Path`** the cookie is applied to (automatic wildcarding):
  `/` matches all pages, `/todos` all pages within `todos`, etc.
- Possible flags:
  - **`Secure`**
  - **`httpOnly`**
  - **`Signed`**



Making cookies more robust

`Secure` flag via HTTP: the cookie will **not** be sent

- **`Secure`** cookies:
  - Ensures that cookies are sent via HTTPS
- **`HttpOnly`** cookies:
  - Cookies are **not accessible** to non-HTTP entities (e.g. client-side **JavaScript**)
  - Minimises the threat of cookie theft
  - Should always be applie**d** **Hash Message Authentication Code** s
- **`Signed`** cookies (appended `HMAC[value]`):
  - Server can check whether the **value** has been **tampered** with by the client `s%3Amonster.TdcGYBnkcvJsd0%2FNcE2L%2Bb8M55geOuAQt48mDZ6RpoU`

Cookie field `Domain`

- **Origin**: **request domain** of the cookie (a cookie is always applicable to its origin server)

`GET http://www.my_site.nl/todos` → `www.my_site.nl`

  - Port or scheme can differ, the received cookie is also applicable to `https://www.my_site.nl:3005`

- A cookie's `Domain` attribute has to cover the **origin domain**
  - If not set, a cookie is only applicable to its origin domain (a cookie from `www.my_site.nl` is not applicable to `my_site.nl`)
  - If set, a cookie is applicable to the domain listed in the attribute and all its **subdomains**

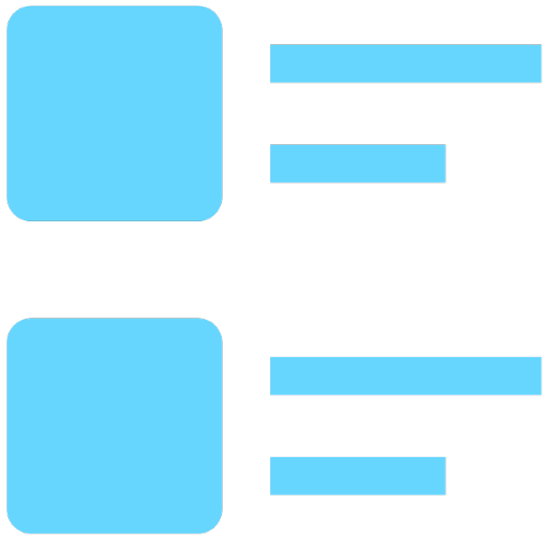
```
GET http://www.my_site.nl/todos
Set-Cookie: name=value; Path=/; Domain=my_site.nl
```

applicable to

```
www.my_site.nl todos.my_site.nl
serverA.admin.todos.my_site.nl
```

Server to client

Client to server



name/value pairs

+

cookie fields

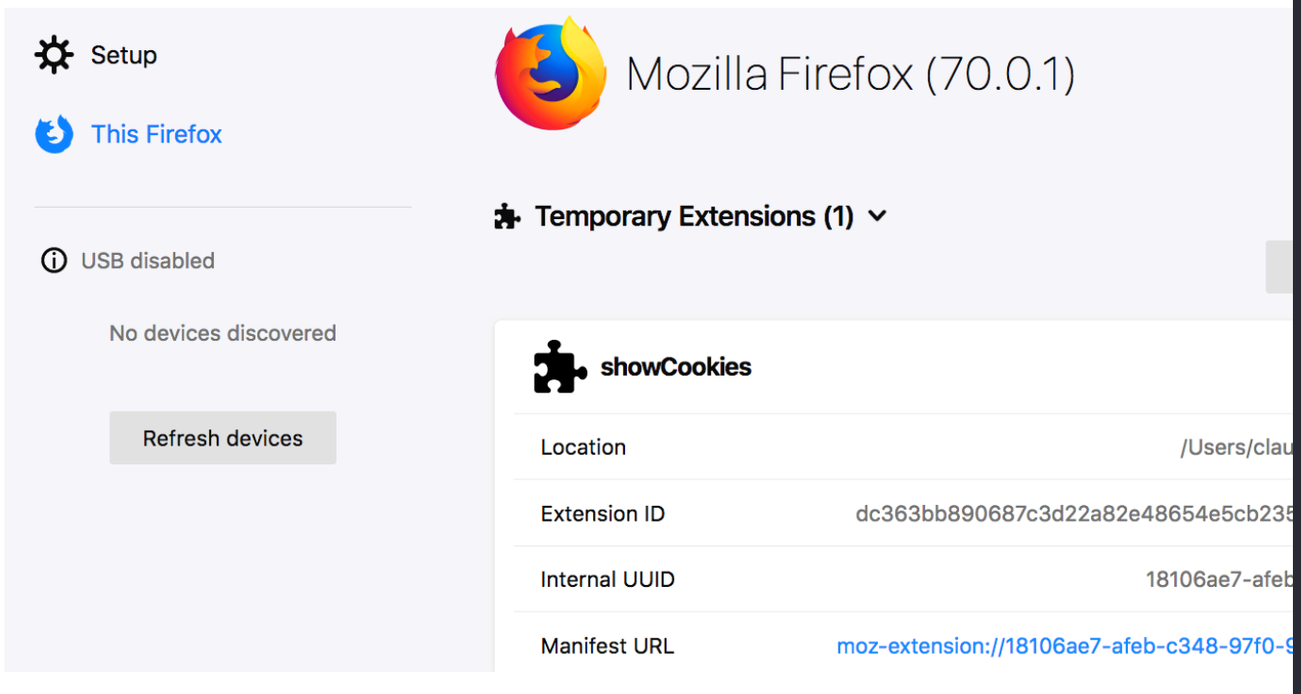
name/value pairs

Excursion …

Extensions (here: Firefox-specific, but very similar for Chrome)

Idea: for every website to visit, see the cookie name/values pairs displayed on the left-hand side of the screen.

manifest.json
showCookies.json



address bar `about:debugging`

```
{
  "manifest_version": 2,
  "name": "showCookies",
  "version": "0.1",
  "description": "Show off cookies.",
  "content_scripts": [
    {
      "matches": [
        "<all_urls>"
      ],
      "js": [
        "showCookies.js"
      ]
    }
  ]
}
```

Extensions

manifest.json
showCookies.json

1. Create a element.
2. Style the .
3. Create a <h2> with the num. of cookies**.
4. Walk over the cookies available through JS and create a <p> element for each.
5. Style each <p>.
6. Append each <p> to the .
7. Append the to <body>.

```
let cookieWindow = document.createElement("div");
cookieWindow.style.position = "absolute";
cookieWindow.style.top = "0px";
cookieWindow.style.left = "0px";
cookieWindow.style.width = "400px";
cookieWindow.style.height = "100%";
cookieWindow.style.opacity = "0.8";
cookieWindow.style.color = "navy";
cookieWindow.style.zIndex = "999";
cookieWindow.style.backgroundColor = "gold";
cookieWindow.style.padding = "10px";
cookieWindow.style.wordWrap = "break-word";

let cookiesArray = document.cookie.split('; ');

let header = document.createElement("h2");
header.textContent = cookiesArray.length + " cookies";
cookieWindow.appendChild(header);

for (let i = 0; i < cookiesArray.length; i++) {
    let cookie = cookiesArray[i].split("=");
    let para = document.createElement("p");
    para.style.padding = "5px";
    para.style.fontSize = "110%";
    if (i % 2 == 0) {
        para.style.backgroundColor = "seashell";
    }
    else {
        para.style.backgroundColor = "mistyrose";
    }
    para.textContent = cookie[0] + " => " + cookie[1];
    cookieWindow.appendChild(para);
}

document.body.appendChild(cookieWindow);
```

app.js — Web-Teaching

EXPLORER

OPEN EDITORS
- app.js demo-code/node-cookies-ex

WEB-TEACHING
- .vscode
  - launch.json
- demo-code
  - balloons-game
  - balloons-wireframes
  - lecture-5-demos
  - node-ajax-ex
  - node-component-ex
  - node-cookies-ex
    - node_modules
    - app.js
    - credentials.js
    - package-lock.json
    - package.json
  - node-ejs-ex
  - node-express-ex
  - node-file-watching-ex
  - node-sessions-ex
  - node-tcp-ex
  - node-url-routing-ex
  - node-web-ex
  - node-websocket-ex
  - README.md
- img
- slides
- .gitignore

OUTLINE

app.js

```
You, 2 months ago | 1 author (You)
1  var express = require("express");      You, 2 months ago • lecture 7 done, apart from third-part
2  var http = require("http");
3  var credentials = require("./credentials");
4  var cookies = require("cookie-parser");
5
6  console.log("credentials: " + credentials.cookieSecret);
7  var app = express();
8  app.use(cookies(credentials.cookieSecret));
9
10 var port = process.argv[2];
11 http.createServer(app).listen(port);
12
13 app.get("/sendMeCookies", function (req, res) {
14     console.log("Handing out cookies");
15     res.cookie("chocolate", "kruemel");
16     res.cookie("signed_chocolate", "monster", { signed: true });
17     res.send();
18 });
19
20 app.get("/listAllCookies", function (req, res) {
21     console.log("++ unsigned ++");
22     console.log(req.cookies);
23     console.log("++ signed ++");
24     console.log(req.signedCookies);
25     res.clearCookie("chocolate");
26     res.send();
27 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
warning: refname 'HEAD' is ambiguous.
warning: refname 'HEAD' is ambiguous.

 2018-11-30 11:04:35  Claudias-MacBook-Air in ~/GitHub/Web-Teaching
± |master → origin ↓29 {5} ✓| →
```

Demo time: node-cookies-ex

```
npm install cookie-parser
```

Cookies in Express

```
1  var express = require("express");
2  var http = require("http");
3  var credentials = require('./credentials.js');
4  var cookies = require("cookie-parser");
5
6  var app = express();
7  app.use(cookies(credentials.cookieSecret));
8  http.createServer(app).listen(port);
9
```

`cookie-parser` middleware

```
10 app.get("/sendMeCookies", function (req, res) {
11     console.log("Handing out cookies");
12     res.cookie("chocolate", "kruemel");
13     res.cookie("signed_choco", "monster", { signed: true});
14     res.send();
15 });
16
17 app.get("/listAllCookies", function (req, res) {
18     console.log("++++ unsigned ++++");
19     console.log(req.cookies);
20     console.log("++++ signed ++++");
21     console.log(req.signedCookies);
22     res.send();
23 });
```

creating cookies

signing a cookie

reading cookies

Accessing and deleting cookies in Express

- **Accessing the value** of a particular key/value pair:
  `var val = req.signedCookies.signed_choco;`

cookie key

- **Deleting** a cookie:
  `res.clearCookie('chocolate');` delete in the **response**!

```
res.clearCookie = function clearCookie(name, options) {
    var opts = merge({ expires: new Date(1), path: '/' }, options);

    return this.clearCookie(name, '', opts);
};
```

Express code base ....

A more pessimistic view

Evercookie

"`evercookie` is a javascript API available that produces **extremely persistent cookies** in a browser. Its goal is to **identify a client** even **after they've removed standard cookies** […]
`evercookie` accomplishes this by storing the cookie data in **several types of storage mechanisms** that are available on the local browser.
Additionally, if `evercookie` has found the user has removed any of the types of cookies in question, it **recreates** them using each mechanism available."

Source: https://github.com/samyk/evercookie

Evercookie

Browser Storage Mechanisms

Client browsers must support as many of the following storage mechanisms as possible in order for Evercookie to be effective.

- Standard HTTP Cookies
- Flash Local Shared Objects
- Silverlight Isolated Storage
- CSS History Knocking
- Storing cookies in HTTP ETags (Backend server required)
- Storing cookies in Web cache (Backend server required)
- HTTP Strict Transport Security (HSTS) Pinning (works in Incognito mode)
- window.name caching
- Internet Explorer userData storage
- HTML5 Session Storage
- HTML5 Local Storage
- HTML5 Global Storage
- HTML5 Database Storage via SQLite
- HTML5 Canvas - Cookie values stored in RGB data of auto-generated, force-cached PNG images (Backend server required)
- HTML5 IndexedDB
- Java JNLP PersistenceService
- Java exploit CVE-2013-0422 - Attempts to escape the applet sandbox and write cookie data directly to the user's hard drive.

Source: https://github.com/samyk/evercookie

First-party cookies are cookies that belong to the same domain that is shown in the browser's address bar.

Third-party cookies

Set-Cookie: x.org

first party

Web server x.org

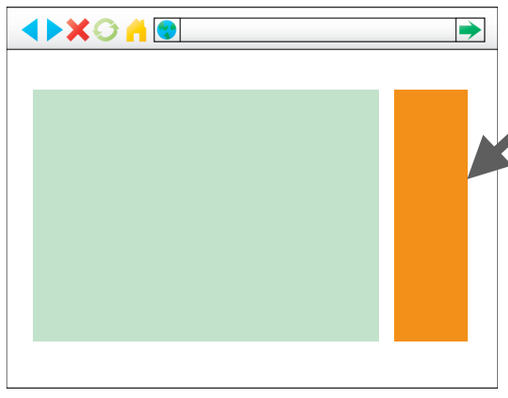
Set-Cookie: ads.agency.com

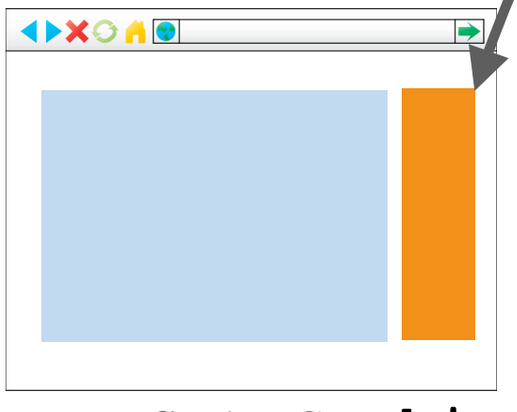
third party

Global Ad Agency
ads.agency.com

served by the same ad agency

Set-Cookie: ads.agency.com





user visits three different websites

based on the cookies a complete user profile can be created

Set-Cookie: ads.agency.com

Browser-based fingerprinting

The rich programming interfaces (APIs) provided by web browsers can be diverted to collect a browser fingerprint. A small number of queries on these interfaces are sufficient to build a fingerprint that is statistically unique and very stable over time. Consequently, the fingerprint can be used to track users. Our work aims at mitigating the risk of browser fingerprinting for users privacy by 'breaking' the stability of a fingerprint over time. We add randomness in the computation of selected browser functions, in order to have them deliver slightly different answers for each browsing session. Randomization is possible thanks to the following properties of browsers implementations: (i) some functions have a nondeterministic specification, but a deterministic implementation; (ii) multimedia functions can be slightly altered without deteriorating user's perception. We present FPRandom, a modified version of Firefox that adds randomness to mitigate the most recent fingerprinting algorithms, namely canvas fingerprinting, AudioContext fingerprinting and the unmasking of browsers through the order of JavaScript properties. We evaluate the effectiveness of FPRandom by testing it against known fingerprinting tests. We also conduct a user study and evaluate the performance overhead of randomization to determine the impact on the user experience.

Paper abstract of https://link.springer.com/chapter/10.1007/978-3-319-62105-0_7

Client-side cookies

Cookies in JavaScript

- Not always necessary to receive cookies from a server
- Cookies can be **set in the browser**
- Canonical use case: remember form input

```
//set TWO(!) cookies
document.cookie = "name1=value1";
document.cookie = "name2=value2; expires=Fri, 14-Jan-2022 12:45:00 GMT";

//delete a cookie by RESETTING the expiration date
document.cookie = "name2=value2; expires=Fri, 24-Jan-2019 12:45:00 GMT";
```

Reading cookies in JavaScript

- `document.cookie["name1"]` does **not work**
- **String** returned by `document.cookie` needs to be parsed

```
let cookiesArray = document.cookie.split('; ');
let cookies = [];

for (let i = 0; i < cookiesArray.length; i++) {
    let cookie = cookiesArray[i].split("=");
    cookies[cookie[0]] = cookie[1];
}
```

Sessions

Introduction (once again)

- **Scenario**: users interact with a web application for a short amount of time (a session)
- **Goals**:
  - Track the user without relying (too much) on cookies
  - Allow large amounts of data to be stored
- **Problem**: without cookies the server cannot tell clients apart
- **Solution**: **hybrid approach** between cookies and server-side saved data

Sessions in one slide

- Cookies are used to store a **single ID** on the **client**
- **Remaining user information** is stored **server-side**

Establishing a session

1. **Client** requests a first page from the server
2. Server creates **unique session ID** and initiates the storage of the session data for that client
3. Server sends back a page with a **cookie** containing the session ID
4. From now on, the client sends **page requests together with the cookie**
5. Server can use the **ID to personalise** the response
6. A **session ends** when no further requests with that session ID come in (timeout)

Sessions in Express with in-memory stores

- Easy to set up in Express
- Same drawback as any in-memory storage: not **persistent** across machine failure
- Middleware component **`express-session`** https://github.com/expressjs/session
- Most common use case is **authentication**

app.js — Web-Teaching

EXPLORER

OPEN EDITORS
- app.js demo-code/node-sessions-ex

WEB-TEACHING
- .vscode
  - launch.json
- demo-code
  - balloons-game
  - balloons-wireframes
  - lecture-5-demos
  - node-ajax-ex
  - node-component-ex
  - node-cookies-ex
  - node-ejs-ex
  - node-express-ex
  - node-file-watching-ex
  - node-sessions-ex
    - app.js
    - credentials.js
    - package.json
  - node-tcp-ex
  - node-url-routing-ex
  - node-web-ex
  - node-websocket-ex
  - README.md
- img
- slides
- .gitignore
- .travis.yml
- Assignment-1.md

OUTLINE

app.js

You, 2 months ago | 1 author (You)

```
1  var express = require("express");      You, 2 months ago • lecture 7 done, apart from third-part
2  var http = require("http");
3  var credentials = require("./credentials");
4  var cookies = require("cookie-parser");
5  var sessions = require("express-session");
6  var app = express();
7
8  app.use(cookies(credentials.cookieSecret));
9  app.use(sessions(credentials.cookieSecret));
10 http.createServer(app).listen(3001);
11
12 app.get("/countMe", function (req, res) {
13     var session = req.session;
14     if (session.views) {
15         session.views++;
16         res.send("You have been here " + session.views + " times (last visit: " + session.lastVi
17         session.lastVisit = new Date().toLocaleDateString();
18     }
19     else {
20         session.views = 1;
21         session.lastVisit = new Date().toLocaleDateString();
22         res.send("This is your first visit!");
23     }
24 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
warning: refname 'HEAD' is ambiguous.
warning: refname 'HEAD' is ambiguous.

 2018-11-30 11:04:35  Claudias-MacBook-Air in ~/GitHub/Web-Teaching
± |master → origin ↓29 {5} ✓| →
```

Demo time: node-sessions-ex

Sessions in Express with memory stores

```
npm install cookie-parser
npm install express-session
```

```
var express = require("express");
var http = require("http");
var credentials = require("./credentials");
var cookies = require("cookie-parser");
var sessions = require("express-session");

var app = express();
app.use(cookies(credentials.cookieSecret));
app.use(sessions(credentials.cookieSecret));
http.createServer(app).listen(3001);

app.get("/countMe", function (req, res) {
    var session = req.session;
    if (session.views) {
        session.views++;
        res.send("You have been here " +
              session.views + " times (last visit: " + session.lastVisit + ")");
        session.lastVisit = new Date().toLocaleDateString();
    }
    else {
        session.views = 1;
        session.lastVisit = new Date().toLocaleDateString();
        res.send("This is your first visit!");
    }
});
```

cookie & session setup

session object available on **`req`** object only

session exists!

session does not yet exist

Third-party authentication

Quora

A place to share knowledge and better understand the world.

Continue with Google

Continue with Facebook

Continue With Email. By signing up you indicate that you agree to Quora's Terms of Service and acknowledge Quora's Privacy Policy.

Login

Email

.....

Forgot Password?

Login

You are now logged out of this browser, but are still logged in with other browsers.
Logout of all browsers.

New: Hindi, Indonesian, and Portuguese

Overview

- **Weakest link** in an authenticated application is the **user's password**
- **Application-based decision**
  - Does the application need authentication?
  - Are cookies/sessions enough?
  - If authentication is needed, should third-party authentication be used?

Third-party authentication

- Authenticating users through popular social Web services (Twitter, Facebook, Google, etc.)
- **Easy** to develop for popular platforms
- **Trusted** social Web platforms **provide authentication**, no need to store passwords or employ particular security measures
- **However**: some users may not use social Web platforms or do not like to hand over their data

OAuth 2.0 Authorization Framework

"The OAuth 2.0 authorization framework enables a **third-party application** to obtain **limited access** to an HTTP service, either **on behalf** of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf."

Source: https://tools.ietf.org/html/rfc6749

OAuth 2.0 roles

- **Resource owner**: entity that grants access to a protected resource

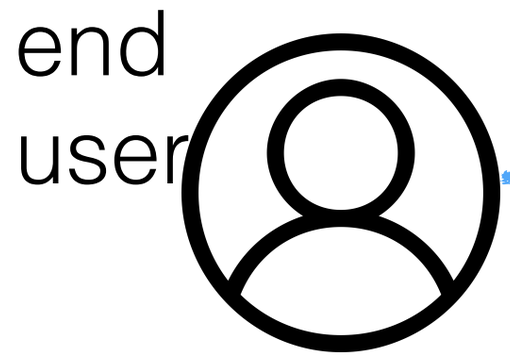
- **Resource server**: server hosting protected resources, capable of accepting and responding to protected resource requests using **access tokens**

a string denoting a specific scope, lifetime and other access attributes

- **Client**: application making protected resource requests on behalf of the resource owner **and with her authorisation**

- **Authorization server**: server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization

OAuth 2.0 roles exemplified



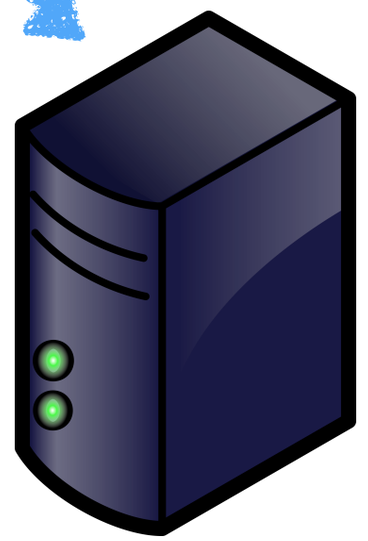
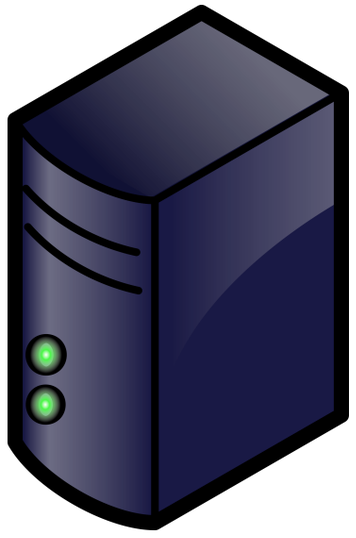


photo sharing service

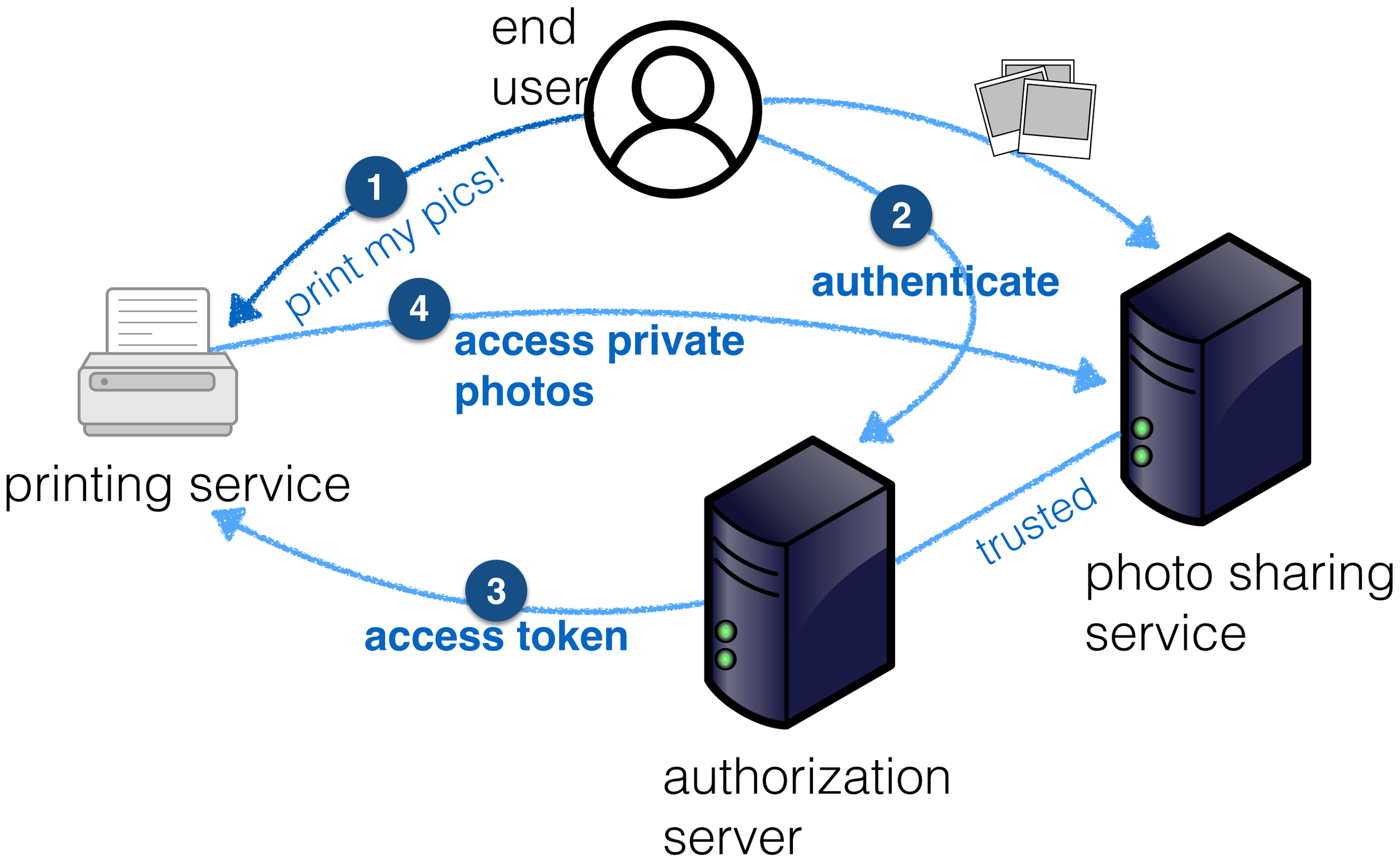
printing service



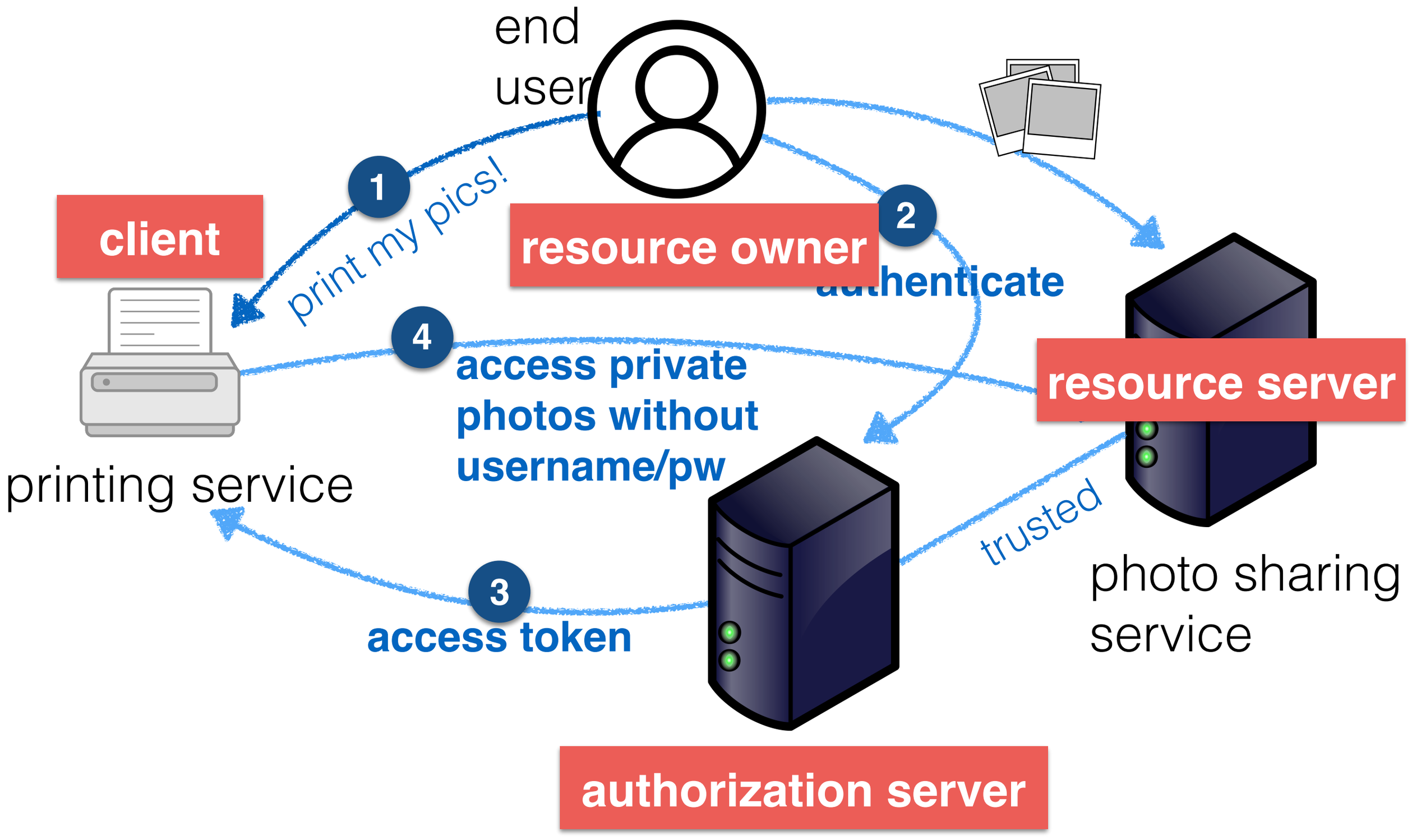
authorization server

Scenario: user wants to grant permission to a printing service to print its private photos stored at a photo-sharing service **without giving away her username/password**.

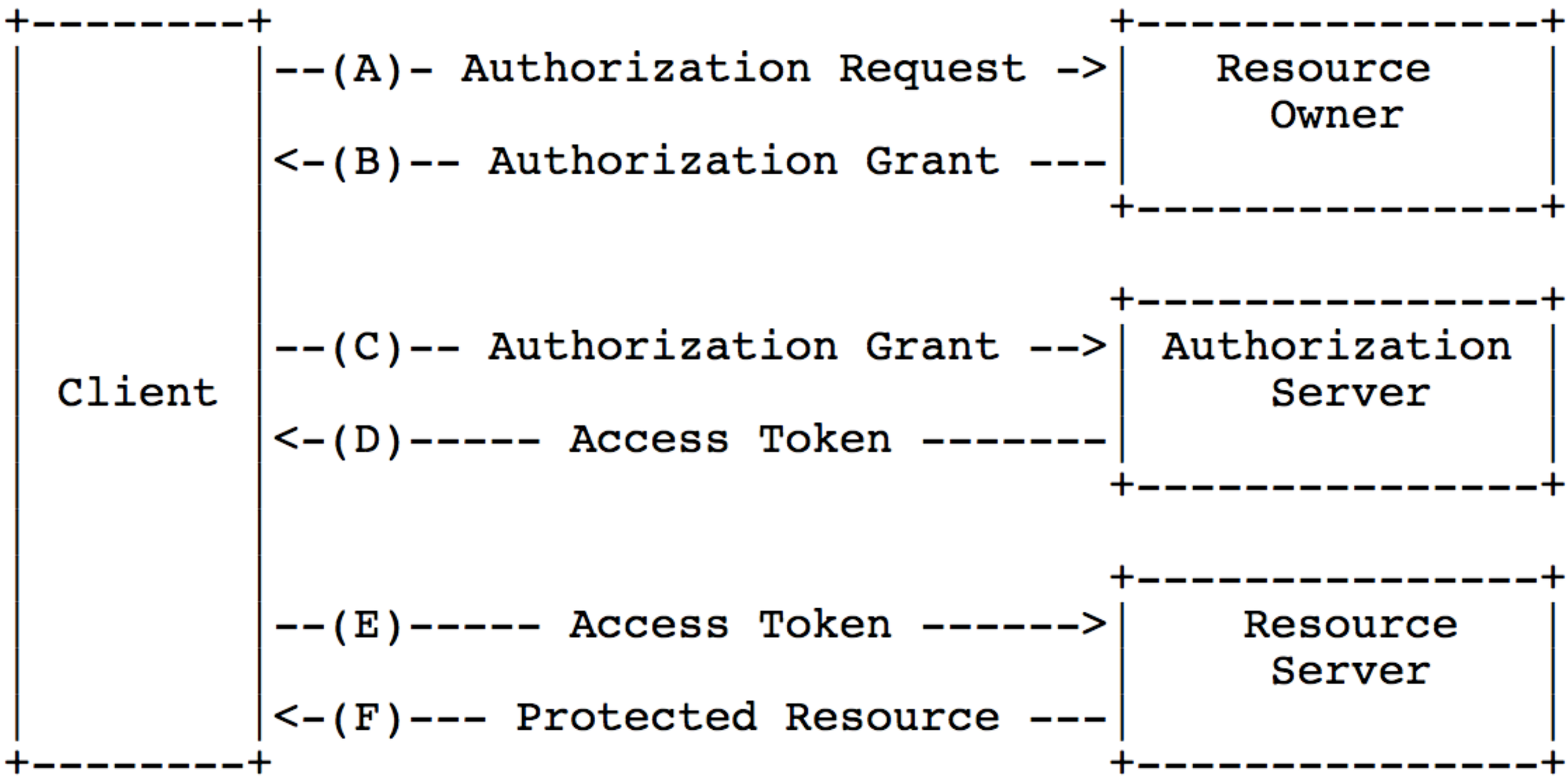
OAuth 2.0 roles exemplified



OAuth 2.0 roles exemplified



Abstract protocol flow



Source: https://tools.ietf.org/html/rfc6749

Third-party authentication with Express

- Express can make use of **`passport`**, one of the most popular **authentication middleware components**
  - 300+ authentication **strategies**
  - Supports OpenID and OAuth
- **`Passport`** hides a lot of the protocol's complexity

```
$ npm install passport
```

```
$ npm install passport-twitter
```

Installing a strategy

http://passportjs.org/

- Work on **Assignment 6.**
- Book yourself an assessment timeslot (one per group).

flickr@courtneyrian