CSC309H1S

Programming on the Web

Winter 2023

Lecture 1: Introduction to the Web

Instructor: Kuei (Jack) Sun

Department of Computer Science
University of Toronto

Course Instructor

- Kuei (Jack) Sun
- Contact Information
  - Use Piazza
    - https://piazza.com/utoronto.ca/winter2023/csc309
    - Sign up to the course to get access
  - By E-mail
    - Personal: sunk@cs.toronto.edu
    - Course Related: csc309-2023-01@cs.toronto.edu
      - Request for lecture-related office hour and special consideration
  - By Calendly
    - https://calendly.com/csc309-2023s
    - Request for project/assignment related office hours and project grading

Course Information

- Course Website on Quercus
  - https://q.utoronto.ca/courses/293527
  - Syllabus
  - Lecture slides/videos/exercises
  - Assignment handouts
  - Project handout
  - Grade posting
- Piazza
  - Course announcement, course discussion
  - Assignment discussion
    - Lab TAs will read and answer relevant posts periodically

Don’t Copy!

- Academic Integrity: Plagiarism and cheating
  - Very serious academic offences
    - All potential cases will be investigated fully
  - All assignments and exercises are to be completed individually
  - Do not submit code for grading that is not your own.
    - If you re-use any code, document the source
      - E.g., hash function from CSC209 A3 starter code, Fall 2019
    - Do not look at others’ code, and do not share your code
    - Do not search for solutions on the web, or use AI-assisted tools, e.g., GitHub Copilot
  - Ask (and answer) questions on Piazza, but don’t add details about your solution
- Exception: term project
  - You may use open-source packages, but they must be clearly referenced

Join or Lead an RSG

- Meet weekly with up to 8 classmates online
- Review and discuss course material
- Prepare for tests and exams
- Get student advice from upper year mentors

Last year, over 3000 students joined a Recognized Study Group (RSG) where they met friends and reached their study goals.

Plan for success this term by joining your RSG today.

Join an RSG today: **uoft.me/recognizedstudygroups**

SIDNEY SMITH COMMONS

@sidneysmithcommons

CCR CO-CURRICULAR RECORD

What we will cover

- How web works
  - Client-server model, Internet, HTTP, browsers
- Static web pages
  - HTML and CSS
- Dynamic website
  - Backend framework, i.e., Django (Python)
- Interactive pages
  - Frontend framework, i.e., React (JavaScript)
- System Administration
  - Deployment (website in production environment)

Term Project

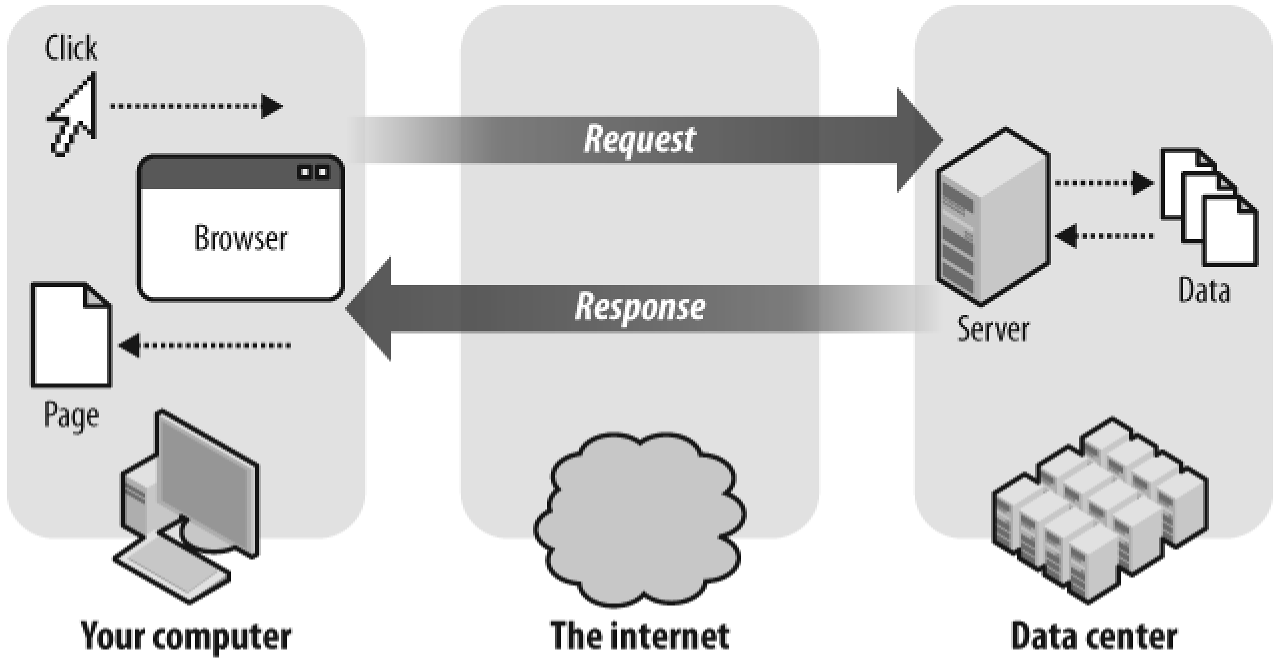
- Group project with up to 3 team members
- Restify
  - Simplified version of Airbnb
  - See project description for detail
- Split into 3 phases
  1. Static Design
  2. Backend API implementation with Django
  3. Frontend implementation with React
- For each milestone, book a grading session with a TA *before deadline*
- Form a group **ASAP** on MarkUs to start planning

Disclaimer

- A lot of material is covered over 12 weeks
- Lectures and tests are focused on knowledge and concepts
  - With some simple coding exercises
- Project requires self-motivated learning
  - Lecture itself is not sufficient to teach every detail of web programming
  - Consult reference manuals
  - Search for answers online
  - Do the assignments for practice
  - Go to mentoring sessions

End User Perspective

1. User enters a web address inside a browser
2. Browser send a request to the server
3. Server processes the request and responds with a web page



https://medium.com/@lokeshchinni123

World Wide Web

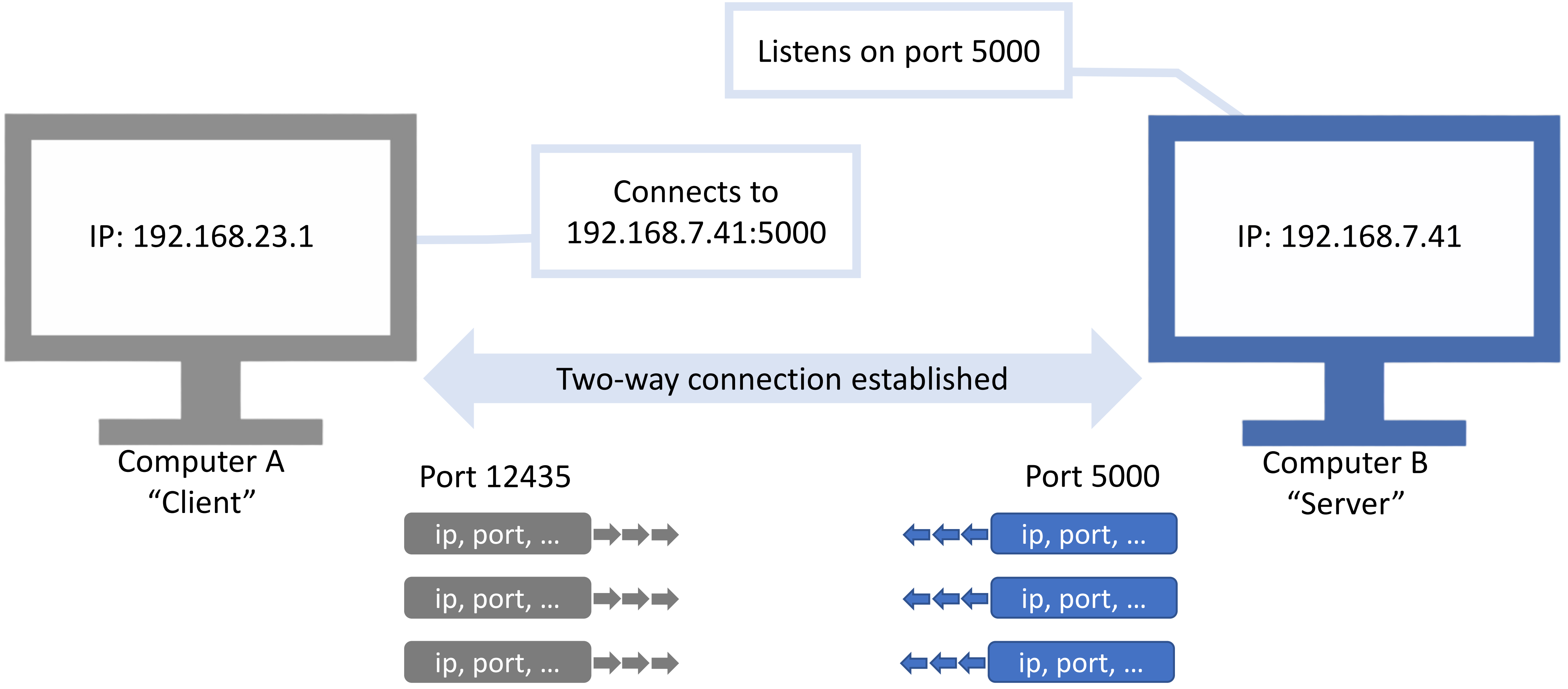
- Also just called “the Web”
- A collection of information and services that can be accessed on local devices through the Internet.
- Internet
  - An interconnected network of computers
  - Can communicate with each other through standardized protocols
- TCP/IP
  - Protocols that provide reliable end-to-end communication between two applications on different computers
- HTTP
  - Protocol for delivery of contents from the Web.

TCP/IP

- IP (Internet Protocol)
  - Identifies computers on the network by assigning a unique IP address
    - E.g., 192.168.7.41
  - Knows how to route data from to the destination computer

- TCP (Transmission Control Protocol)
  - Allows multiple *virtual* connections to share a single physical IP address
  - Each connection is identified by a unique port number
    - E.g., port 80
  - Deals with unreliable nature of data transmission over network

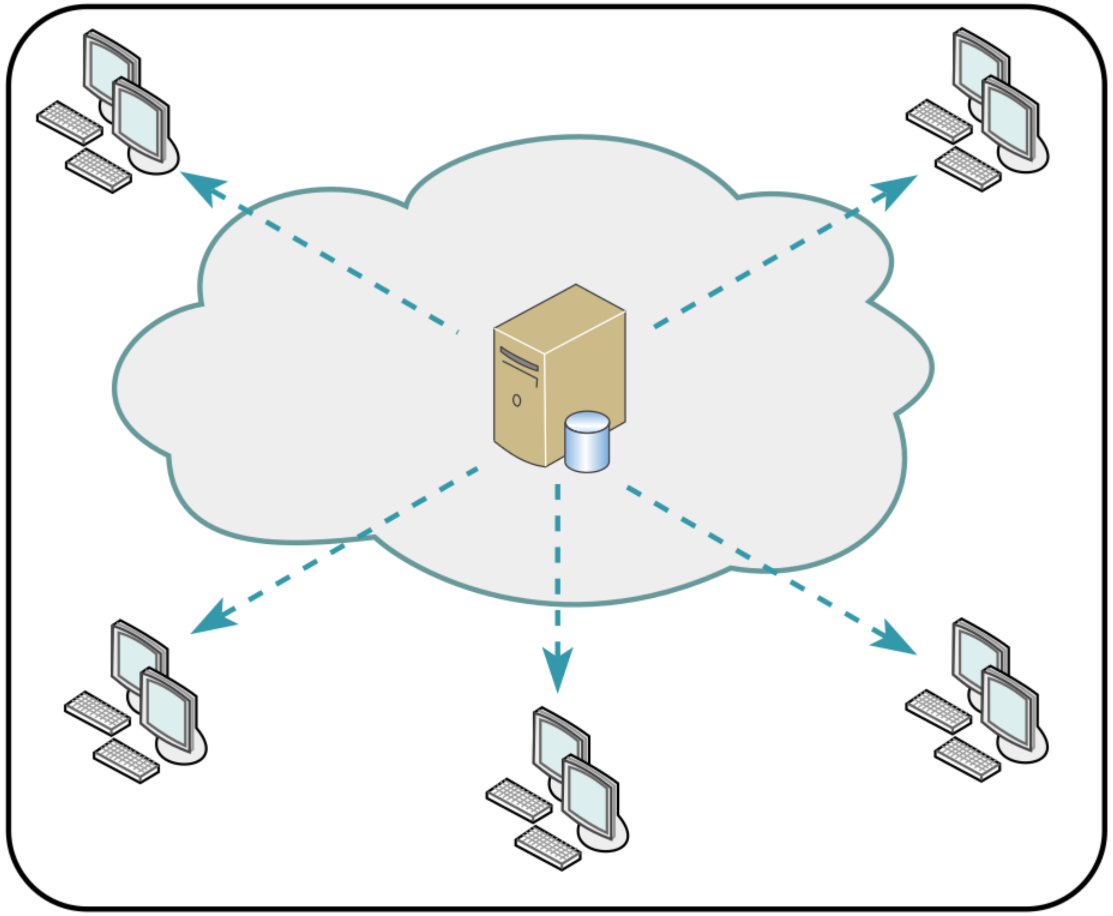
How computers talk to each other?

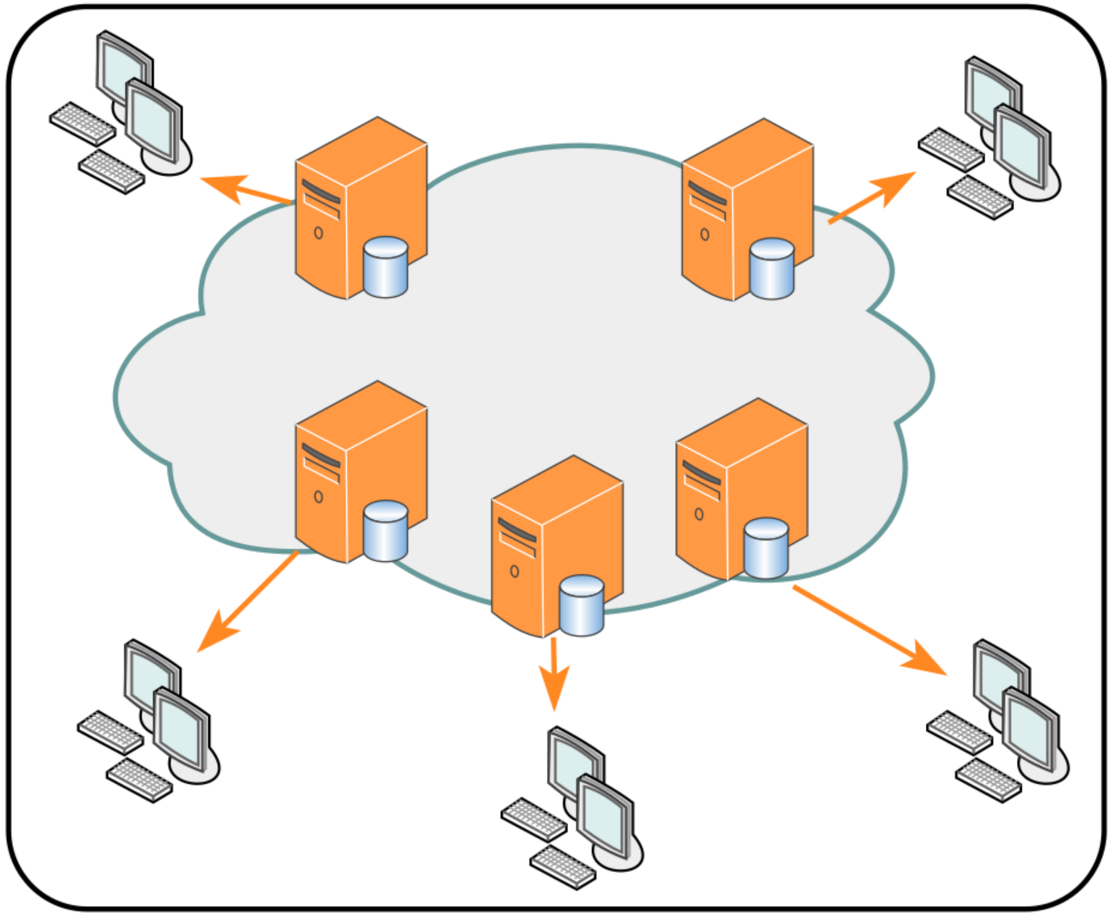


Domains

- IP addresses are hard to remember
- It is possible to move websites elsewhere
- Some websites may be hosted on multiple physical machines
  - Content delivery network (CDN)
- We want an easier way to remember addresses
  - Also want an easy way to remap websites to different IP addresses
- Domain Name
  - Maps an easy-to-remember name to IP address(es)
  - www.google.com → 142.251.41.78
  - Clients must *resolve* the domain before making a connection

Content Delivery Network





Domain Name System

- A collection of mappings from domain names to their IP addresses
  - Analogy: Phone book
- Problem: how do we find the DNS server?

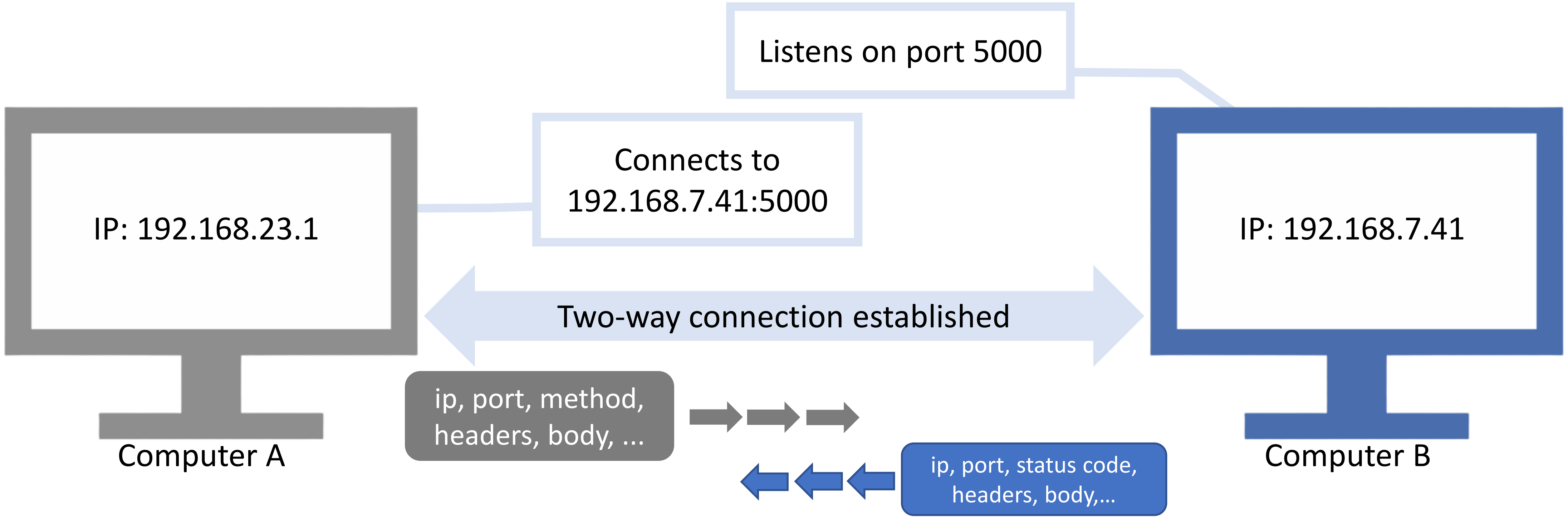
- Manually assigned by system administrator
  - E.g., 8.8.8.8 (Google's Public DNS)
- Automatically configured when computer connects to Internet
  - Computer sends a broadcast message to everyone on the local network
  - The DHCP server is responsible for assigning an IP address to the computer
    - It would also respond with the IP address of a DNS server

Hypertext Transfer Protocol

- A protocol for distributing and accessing hypertext documents
  - Hypertext is text displayed on electronic devices, e.g., monitor
- Built on top of TCP/IP
- Human readable protocol
- HTTP servers typically listen on port 80
- HTTPS (HTTP Secure)
  - Messages are encrypted for security purposes
    - Protects against eavesdropping and tampering
  - Used by 81.3% of all public websites

Stateless Protocol

- HTTP is a *stateless* protocol
- HTTP servers do not remember previous interaction with their clients



Listens on port 5000
Connects to 192.168.7.41:5000
IP: 192.168.23.1
IP: 192.168.7.41
Two-way connection established
ip, port, method, headers, body, ...
ip, port, status code, headers, body,...
Computer A
Computer B

Statefulness

- A stateful service reacts *differently* to the same input
- Server must track the states all open connections
- Example: money transfer
  1. Enter account password
  2. Enter amount and recipient
  3. Confirm transfer
- Online banking service requires knowing that step 1 was successful
  - A stateful server remembers this on the server-side (the bank)
  - A stateless server gives the client a cookie to be passed back later
    - The client *reminds* the server of the previous step

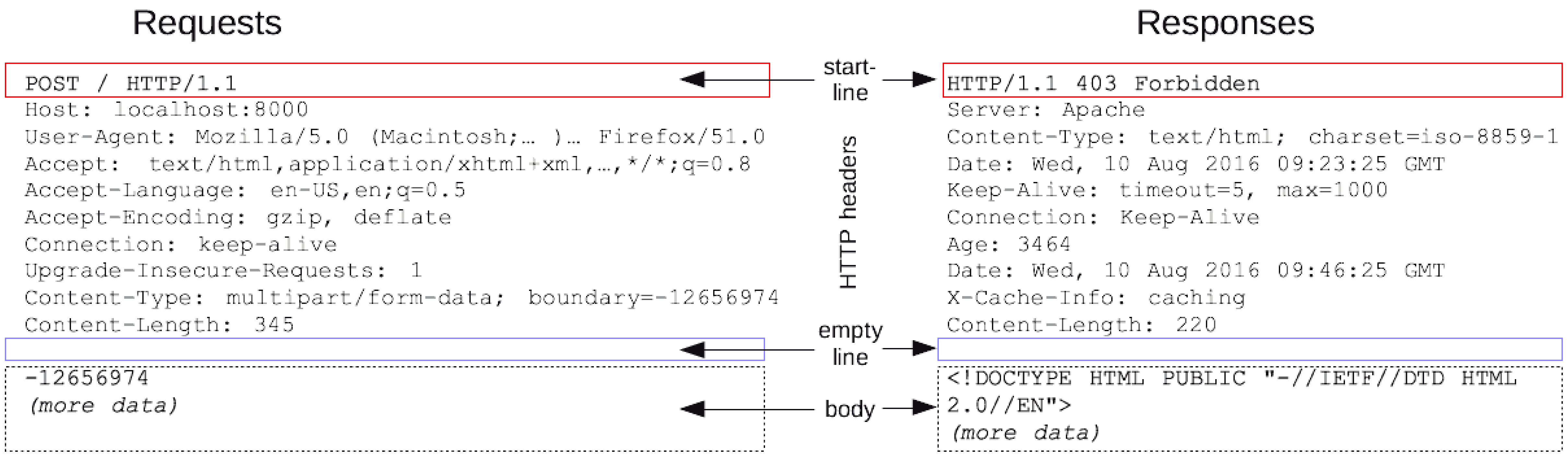
Statefulness

- Stateful service
  - Requires server to keep information about a session (interaction with client)
  - More complicated to design and implement
  - Server crash or power outage would result in loss of session states
  - Difficult to scale (work smoothly with increased number of users)
- Stateless service
  - Does not require server to remember session states
  - Simple to design and implement
  - Server outage does not result in loss of session states
  - Easier to scale and optimize
    - E.g., by caching responses

HTTP Message

- Components of an HTTP Request
  - Method: describes what you want to do
  - Path: specifies which resource you want to access
  - Header: describes various settings and client environment
  - Body: additional data to be sent to server
- Components of an HTTP Response
  - Response code: describes the outcome of the request
  - Header: describes various settings and server environment
  - Body: data from the server (usually the hypertext of the web page)

HTTP Message



Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept:  text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language:  en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests:  1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

Responses

```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

HTTP Methods

- POST
  - Create a new resource
- GET (most used)
  - Read information about a resource
- PUT
  - Replace a resource
- PATCH
  - Modify a resource
- DELETE
  - Delete a resource

Response Code

- Success: 200 – 299
  - 200 OK, 201 Created
- Redirection: 300 – 399
  - Instructs user to check out a different web address
  - 301 Moved Permanently
- Client error: 400 – 499
  - 404 Not Found, 400 Bad Request, 403 Permission Denied
- Server error: 500 – 599
  - 500 Internal Server Error, 502 Bad Gateway

Uniform Resource Locator

- A string to reference a web resource and how to retrieve it
- Format of a hyperlink for navigating through hypertext documents
- Example:
  - https://www.utoronto.ca/current-students

    protocol to use   domain name   resource name

- URL encoding
  - Some characters are not safe in documents where URLs may be used
  - Escaped using *percent encoding*: e.g., space is converted to %20

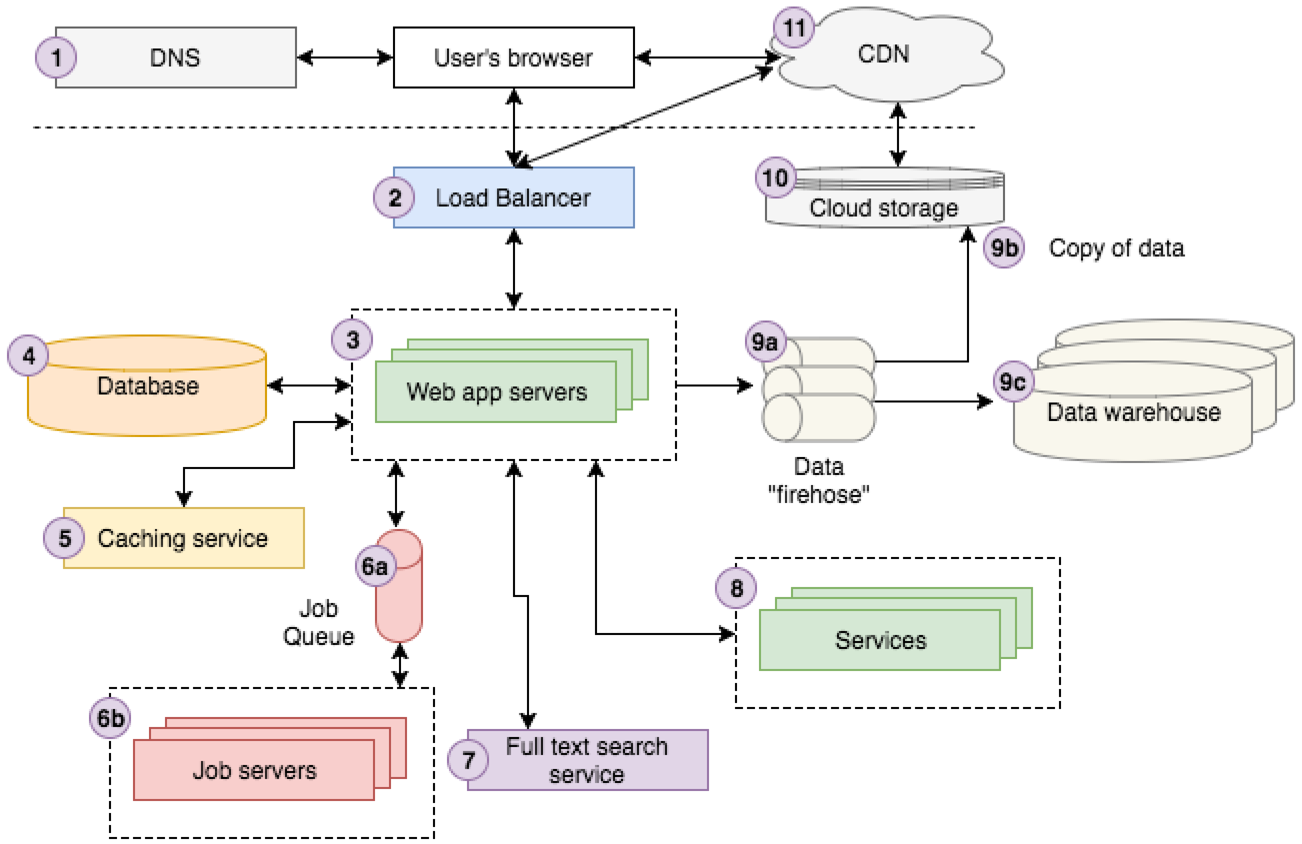
Web Browser

- A client-side application that takes an URL and retrieves a web page
  - Using the HTTP/HTTPS protocol over TCP/IP
- Web pages are typically written in HTML
  - Hypertext Markup Language
- A web browser *renders* the hypertext to display formatted contet
- Popular modern browsers

Safari
Apple
Firefox
Mozilla
Chrome
Google
Edge new
Microsoft
Opera
Opera Software

Modern Web Architecture

- Contains many components, each used for different purposes



https://medium.com/storyblocks-engineering/web-architecture-101-a3224e126947

Summary

- Web server listens on a specific port
    - Client(s) connect to IP address and port number
- DNS translates domain names to IP addresses
    - Users can refer to websites by domain name rather than IP address
- HTTP protocol
    - Stateless
    - Client sends a request and server replies with a response
- HTTP response body is usually in HTML format
    - Browsers understand this format and renders accordingly