CSCI 165
Introduction to the Internet and the World Wide Web
Lecture 1: The Internet 2

Jetic Gū

2024 Spring Semester (S1)

Overview

- Focus: Course Introduction

- Architecture: Computer Network, WWW

- Core Ideas:

  1. Local Area Networks vs Global Network

  2. The WWW

What is the Internet?

- "Interconnection of Networks"

- Decentralised at Gateway level, End-users connect to Gateways to get "Online"

- Routing Problem

- IP Address

Why does my IP address look weird?

- Your home has a router, which provides Wi-Fi
  IP address: **43.10.X.X**; **192.168.1.100**

  - Your laptop
    192.168.1.101

  - Your smartphone
    192.168.1.102

  - Lisa's smartphone
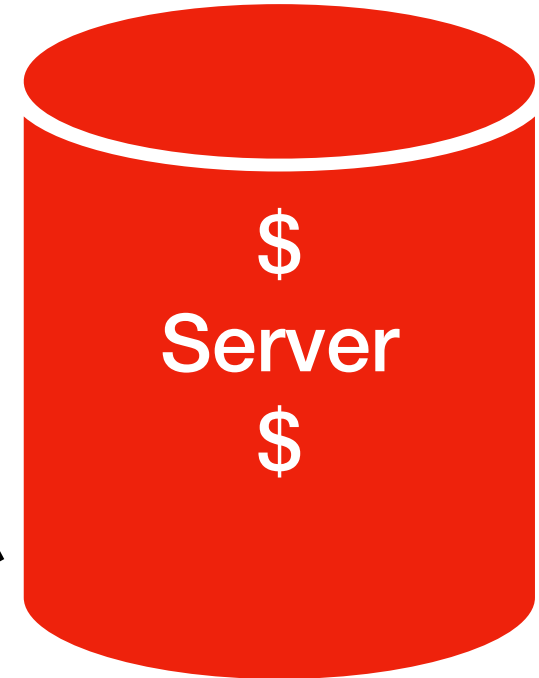   **192.168.1.103**

ISP

Router

Technical
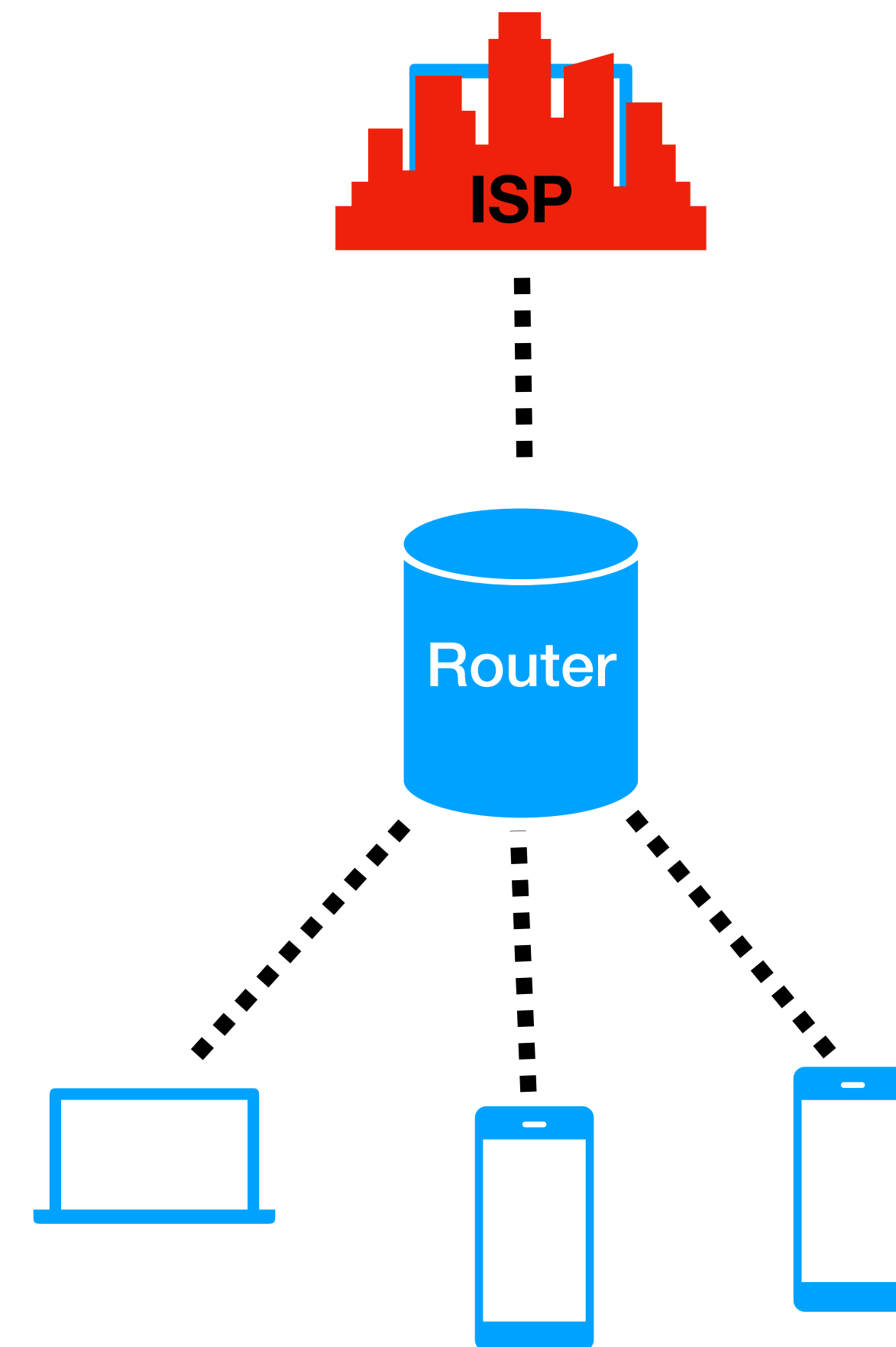
# Why does my IP address look weird?

- Devices sends **Request**s, servers **Respond**

  - A **Response** is sent back the way it came

  - A **Request** requires a destination **IP address**

ISP

Router

Technical

Why does my IP address look weird?

$
Server
$
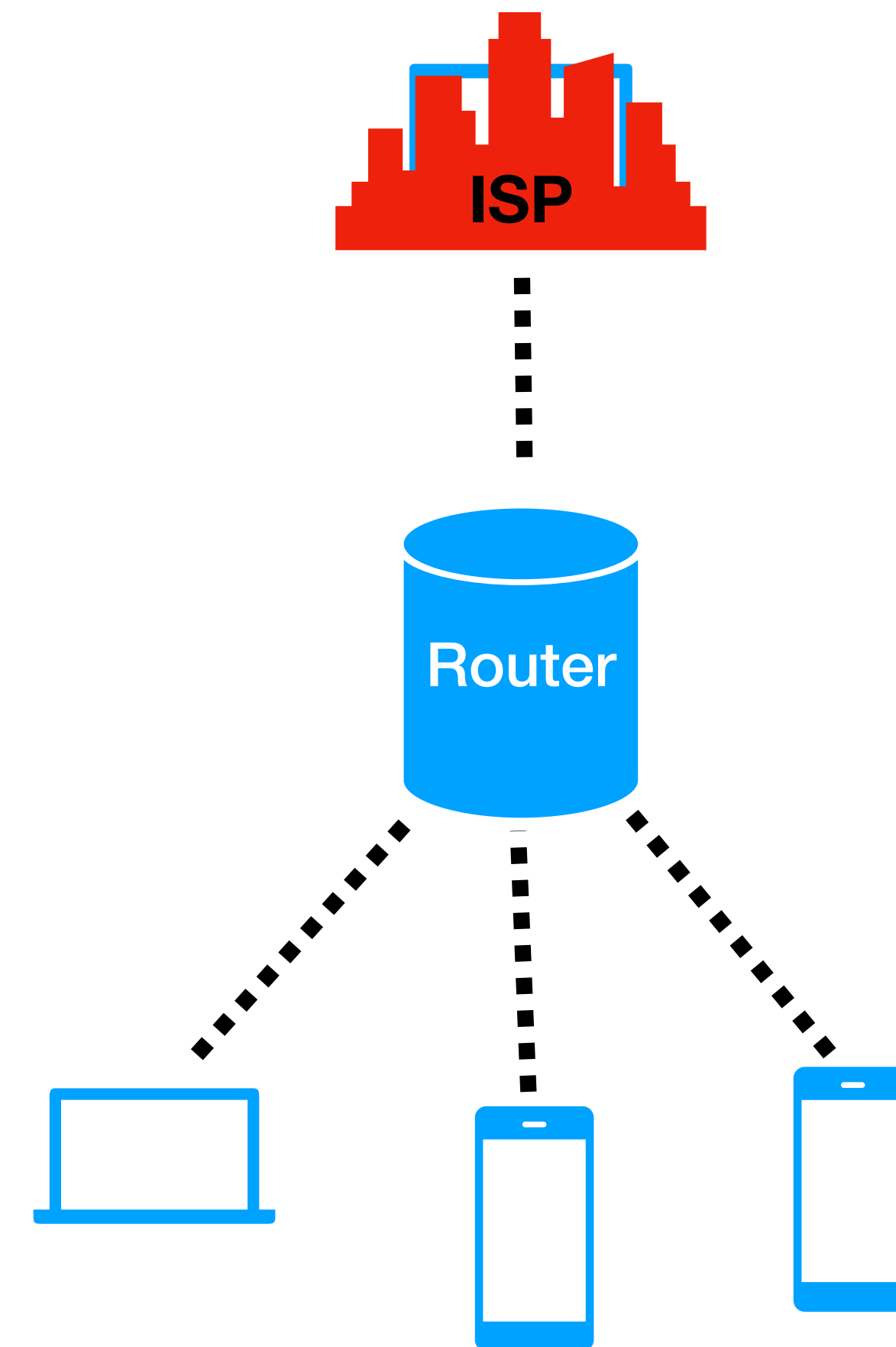
- The router, and the devices connected to it forms a **local area network**

  - **Requests** for services are sent to the Router first
    Routers sends it to the Gateway **on your behalf**

  - **Responses** are sent back to the Router first
    Routers then sends it to the original sender

  - Who can send requests to whom?
    Device -> **Local device**
    Router -> **Server** (Public IP)
    Device(Local) -> **Router** -> **Server** (Public)
    Server -> **Router**
    Server >< **Device**
    Device >< **Remote device**

ISP

Router
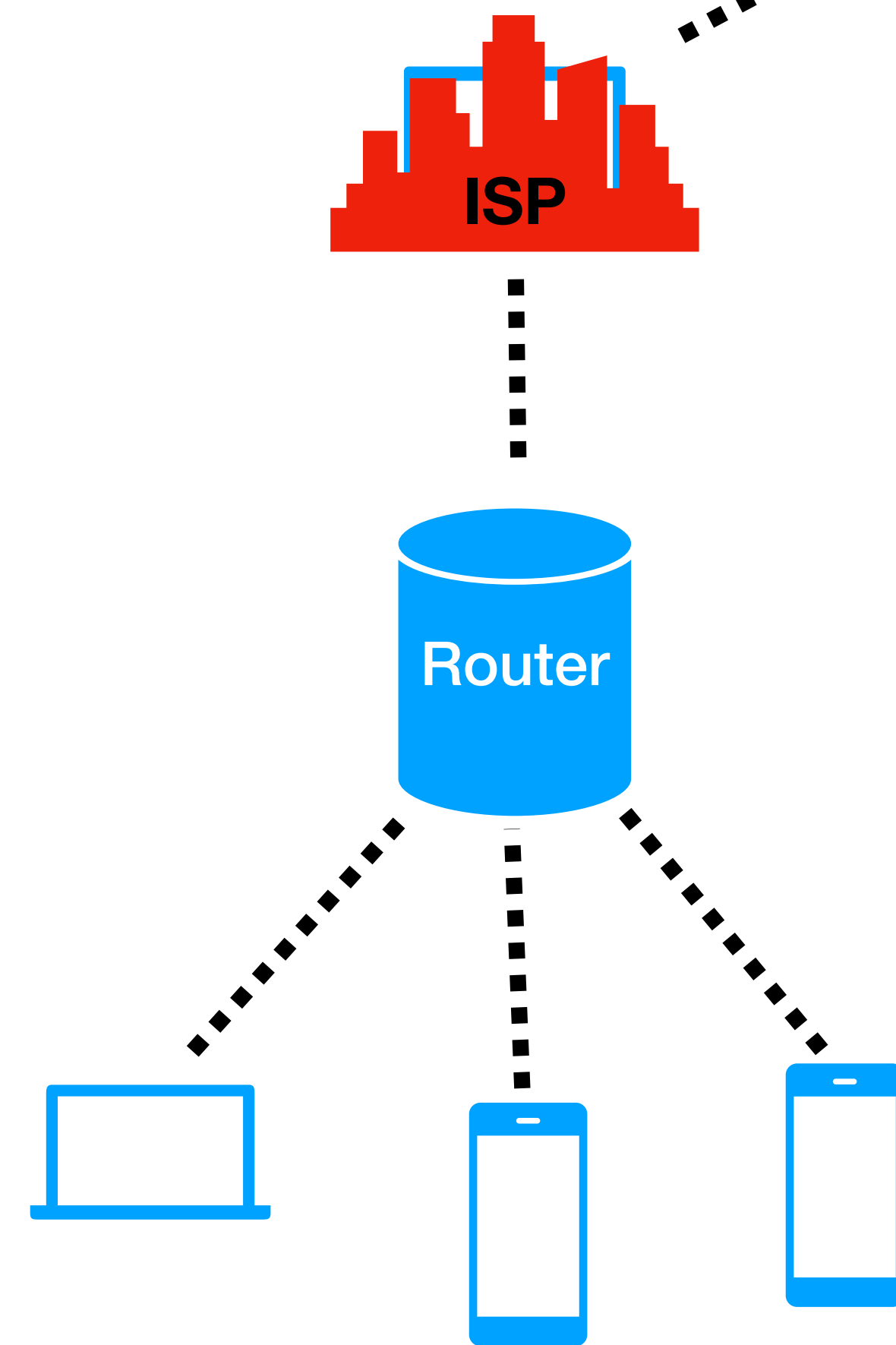
Technical

What is the WWW?

- WWW
  One application (**Web browser**), access countless services (**Websites**)

- Before the World-Wide Web

  - Each internet service you want to access
    1. **Request** a software through mail or telephone order
    2. Service provider **sends you a floppy disk** with software
    3. Install that software, so you can use that **one service**, and one only

Concept

The URL

https://jetic.org/kurs/csci-165

- URL: Uniform Resource Locator

  - You type this to the address bar, press enter, and a **Request** is sent

  - **Response**: webpage

  - What's in the URL?

Concept

The URL

https://jetic.org/kurs/csci-165

- Protocols

  - HTTP: HyperText Transport Protocol

  - HTTPS: HyperText Transport Protocol (Encrypted)

  - others: FTP, FTPS, SMB, etc.

The URL

https://jetic.org/kurs/csci-165

- The Domain

  - A domain is kinda like a **nickname** for the server's **IP address**

  - jetic.org is translated to: 139.162.15.171

  - Who provides this translation service? **A DNS server** (Domain Name System) e.g. Google's DNS server: **8.8.8.8**

Technical

The URL
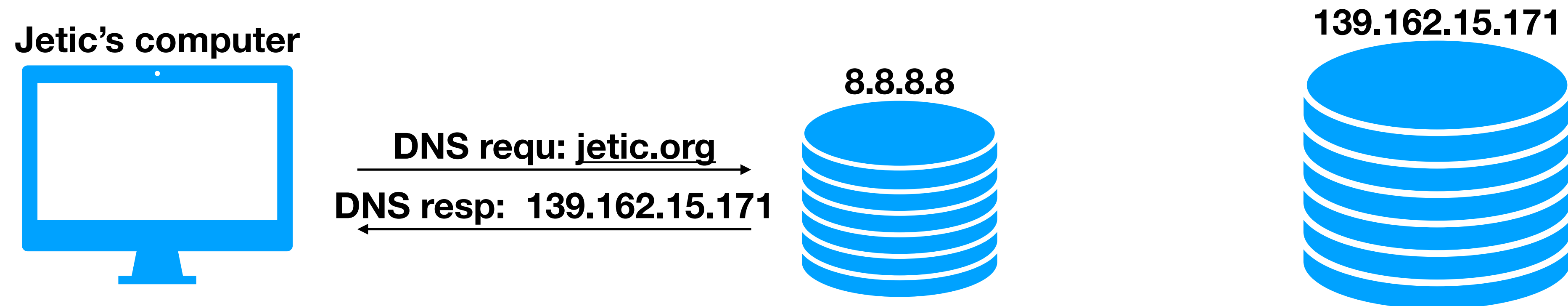
https://jetic.org/kurs/csci-165

Subdirectory

- Your browser will send a **Request** to the DNS server specified in your Network settings (provided by the gateway, router, or manually specified) with the **domain name**
The DNS server **Responds** with the **IP address** of the domain

- Your browser sends an HTTPS **Request** to the **IP address** of the domain
My server **Responds** with the webpage, specified by the **Subdirectory**
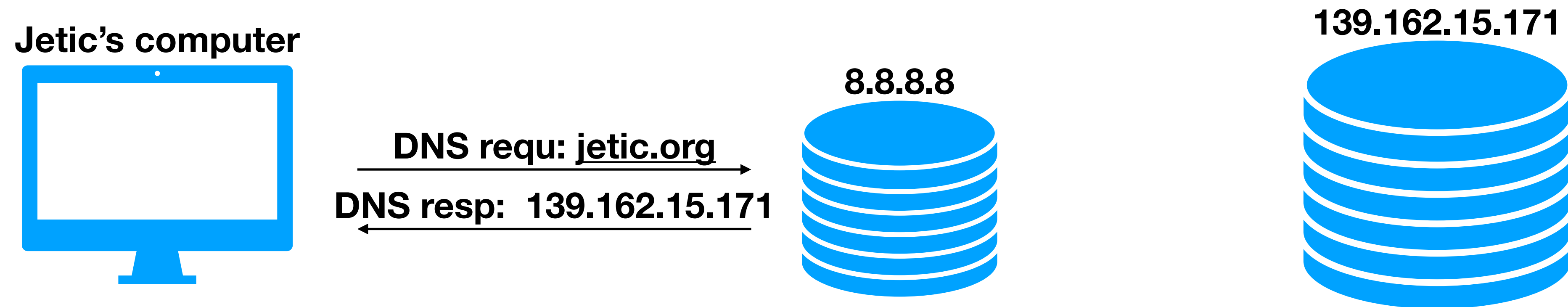
Technical

Server and Client

- In the context of WWW

  - Client: Web browser, sends **Request**s to remote servers
    FireFox, Chrome, Edge, Safari, etc.
    Knows the server's IP address (through DNS)

  - Server: Web server, receives **Request**s and **Respond**s with webpages
    Apache2, IIS, Nginx, etc.
    Knows the sender's gateway's IP address

  - Communications protocol
    TCP/IP (lower), **HTTP/HTTPS** (upper)

    - Can transfer any file formats, but most prominently text-based files
      HTML (webpages), **CSS** (style sheets), **Javascript** (coding scripts)

Concept

Server and Client

Jetic's computer

139.162.15.171

8.8.8.8

DNS requ: **jetic.org** →

← DNS resp: **139.162.15.171**

- How does Jetic access his own website?

  - Jetic opens a browser, types in https://jetic.org

  - jetic.org is sent to a DNS server, the DNS server sends back IP address

  - DNS Requests and Responses are transmitted through **TCP/IP**

Technical

Server and Client

Jetic's computer

139.162.15.171

8.8.8.8

DNS requ: **jetic.org** →

← DNS resp: **139.162.15.171**

- Does the same Domain name always result in the same IP address?

  - Some yes: jetic.org has only one IP address

  - Some no: Google has dozens, each a different data centre, serving Clients from different regions (Americas, EU, Asian, Australia, etc.)

Concept

Server and Client

Jetic's computer

139.162.15.171

HTTPS request: `GET https://jetic.org`

- How does Jetic access his own website?

  - The browser sends an HTTPS Request to 139.162.15.171

  - the Request contains: source Port, destination Port (443), entire URL, etc.

Technical
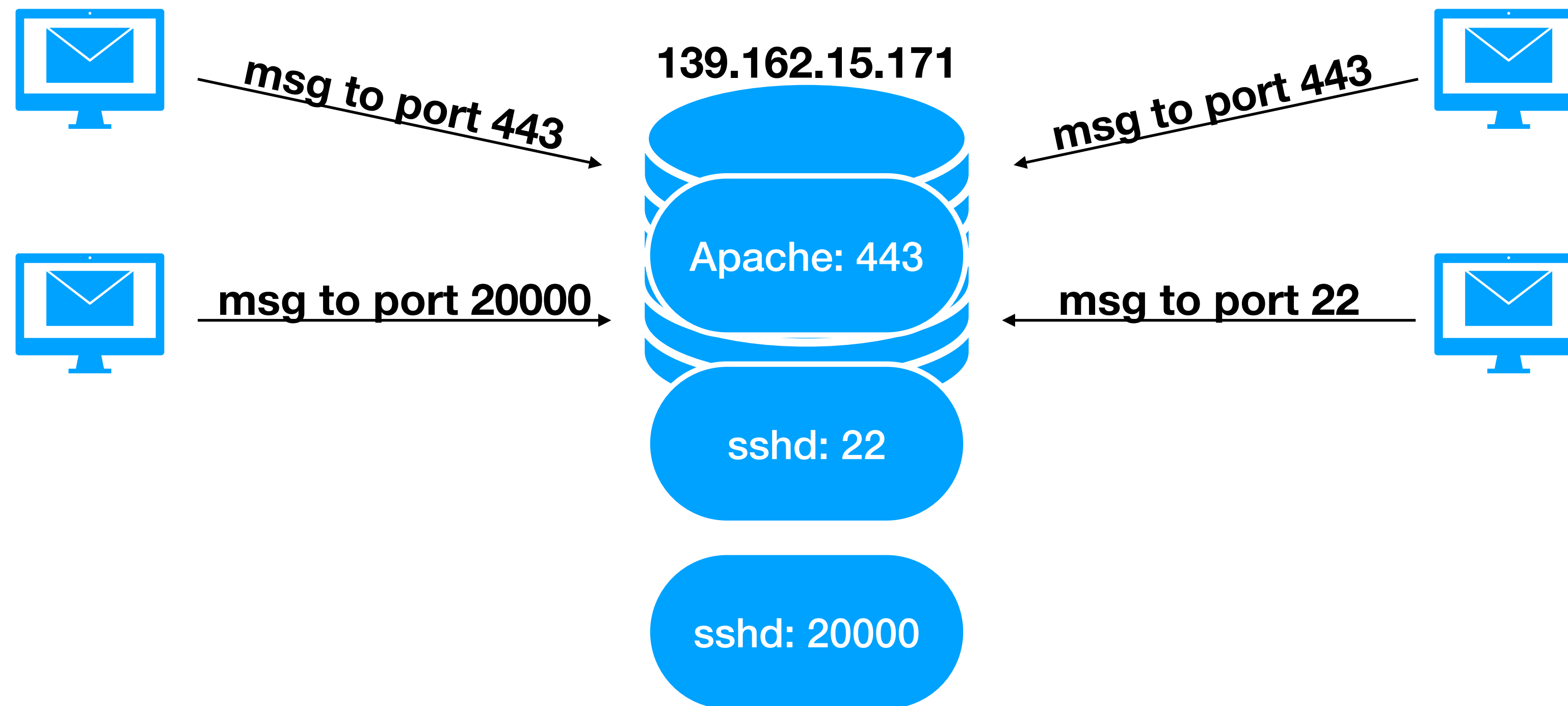
Server and Client

Jetic's computer

139.162.15.171

HTTPS request: `GET https://jetic.org`

- What are ports?

  - Virtual places within an OS where network connections starts and ends

  - A port can be used to send and receive messages

  - A port can only be used/monitored/listened by **ONE** applications at a time

Concept
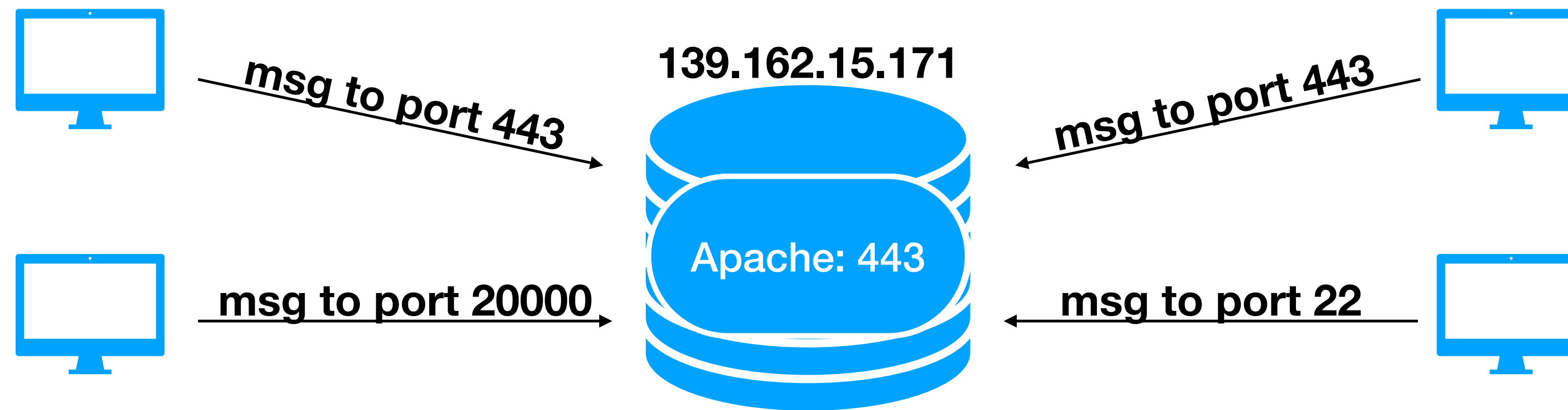
Server and Client

- HTTPS port

  - Port number 443

  - jetic.org uses Apache to process all HTTPS requests, apache2 listens to incoming TCP/IP Requests

Technical

Server and Client

139.162.15.171

msg to port 443

msg to port 443

Apache: 443

msg to port 20000

msg to port 22

sshd: 22

sshd: 20000

• Different applications can handle messages destined for different ports

• Applications can spawn subprocess and create new ports to dedicatedly serve requests

Technical

Server and Client

Jetic's computer

139.162.15.171

HTTPS request: `GET https://jetic.org` →

← **HTTPS response:** `<html>...</html>, code 200`

- How does Jetic access his own website?

  - The server's Apache receives the **Request**, generates the appropriate **content**, sends it back, along with a status code **200**

  - Usually the content is in **HTML (Hypertext Markup Language)** if the request is for a webpage, but can be any other file types

Concept

Server and Client

Jetic's computer

139.162.15.171

HTTPS request: `GET` `https://jetic.org`

HTTPS response: `<html>…</html>, code 200`
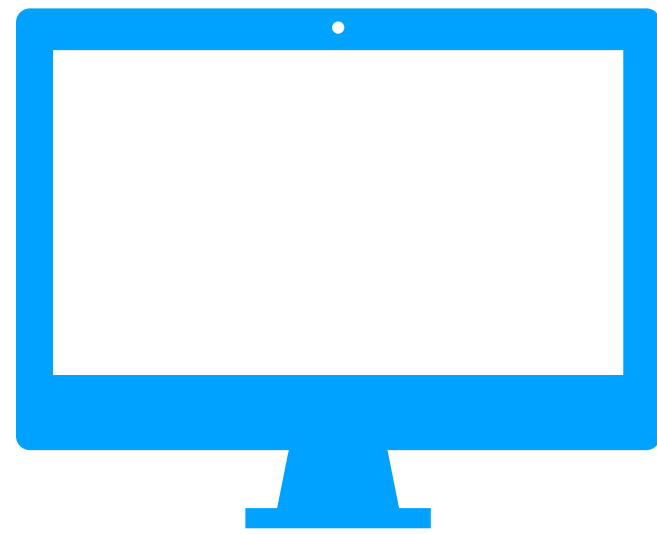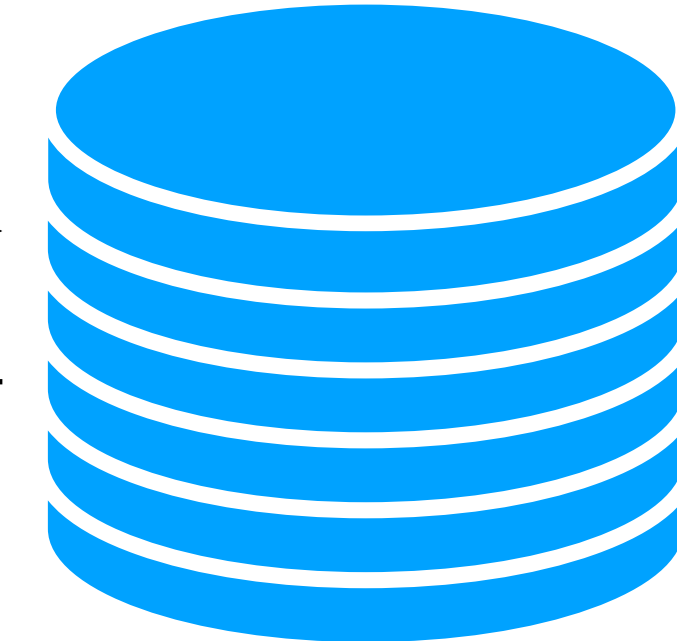
- **Request** methods

  - **GET**: requests for file/webpage, followed by URL

  - **PUT**: sends server texts, in HTML form format
    Used to send server information, such as userid, your password, text for your twitter post, etc.

  - **POST**: sends server resources, such as files, pictures, etc.

Technical

Server and Client

Jetic's computer

139.162.15.171

HTTPS request: `GET https://jetic.org`

HTTPS response: `<html>…</html>, code 200`
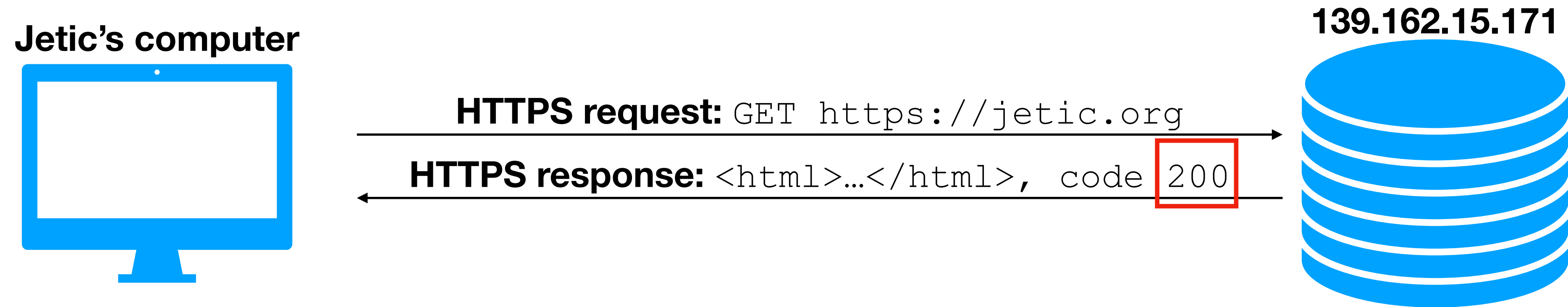
- **Response** status codes (100-599):

  - **200**: <u>successful response</u>, everything is OK

  - **404**: <u>client error response</u>, not found

  - **500**: <u>server error response</u>, internal server error

Technical

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

Server and Client

Pictures

CSS style sheets

Javascript

etc...

Jetic's computer

```
<html>
…
</html>
```

139.162.15.171
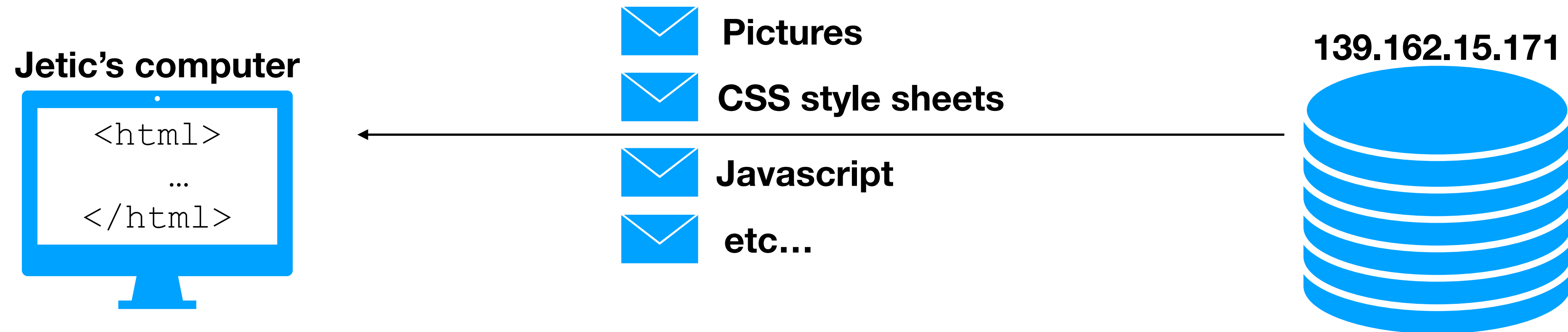
- How does Jetic access his own website?

  - The browser receives the HTML code, start executing it

  - Webpages often **require other resources**, such as **CSS** and **Javascript**s
    These are **Request**ed and **Respond**ed using HTTPS

  - Finally the webpage gets Rendered by the browser, and you can view the content

Concept

Server and Client

- This is a complete list of all resources loaded for jetic.org
  56 different resources

  - 3 HTML documents

  - 21 CSS style sheets

  - 9 Images in Jpg and PNG format

  - 8 fonts in woff2 format

  - 2 vector images in SVG format

  - 13 javascript files

Technical

Developer Tool on Your Browser

- Every major browser has a set of Developer Tools, allowing you to

  - See all downloaded resources for a webpage

  - View individual resources for a webpage

  - Access the Javascript console

  - Inspect HTML elements

  - Monitor ongoing network traffic timeline

  - etc.

Hands On

Developer Tool on Your Browser

P3
Dev

- Your task today

  - Visit one website you often visit

  - Checkout its resources

    - which ones are received successfully? Which ones are not?

    - identify the different types of resources (HTML, CSS, Javascript, Fonts, Images, etc.)

Hands On