COSC 360: Web Development

Course Details

Course slides are based on the resources from the text:

Fundamentals of Web Development. 2nd ed

Authors: Connolly & Hoar

Course information

Instructor:	Matthew Fritter
Office Location	SCI 104
Email	fritter@mail.ubc.ca
Credit Hours	3.0
Presentation format	Lecture 7.5 hrs/wk, Lab 4 hrs/wk
Prerequisite:	Score of 60% or higher in COSC 121 and COSC 304, and third-year standing

- Course website:
  - Canvas– Check frequently

Policies

- ***See Course Outline***

How to Pass the Course

The most important things to do to pass this course:

- Attend class
- Attend labs
- **Participate is discussions and exercises**
- **Take notes**
- **Ask questions**
- Do the assignments early as they are designed to help re-enforce the lecture materials
- Keep up with reading and review

My Goals

- **My Goals for this course:**
  - Prepare students with the necessary techniques and skills required to succeed in modern web development
  - Reinforce theoretical knowledge of web development with practical applications and hands-on development work
  - Do so in a courteous and professional manner and make myself available to students as a resource

Course Objectives

- **In this course students will:**
  - Develop a technical understanding of the Internet environment and the structure of the World Wide Web
  - Gain a solid foundation of modern front end and backend development and integration strategies
  - Technical skill will focus on the design and implementation of effective web sites
  - Work will be conducted on an ongoing project, providing the
  - Opportunity for knowledge and skill synthesis resulting in a practical and functional showcase of acquired skills

Required Materials

- iClicker

- Fundamentals of Web Development. 2nd ed., Connolly and Hoar, 2018.

Course Evaluation

iClickers	5%	
Midterm #1	20%	Tentative – July 23rd, 2019
Labs/Assignment/Quizzes	35%	
Final	40%	During Exam period, TBA
Total	100%	

Major Topics

- Network Basics
- HTML Basics
- CSS Style, Selectors, Box Model
- HTML Tables and Forms
- Advanced CSS layouts and Frameworks
- Client Side Scripting
- Media Formats
- Security
- Server Side Scripting
- Database Connectivity with PHP
- Error Handling and Input Validation
- Managing State
- Data Interchange
- Content Management Systems

COSC 360: Topic 1: How the Web Works

Chapter 1

1. A Complicated Ecosystem
2. Definitions and History
3. The Client-Server Model
4. Where is the Internet?
5. Summary

A Complicated Ecosystem

Definitions and History

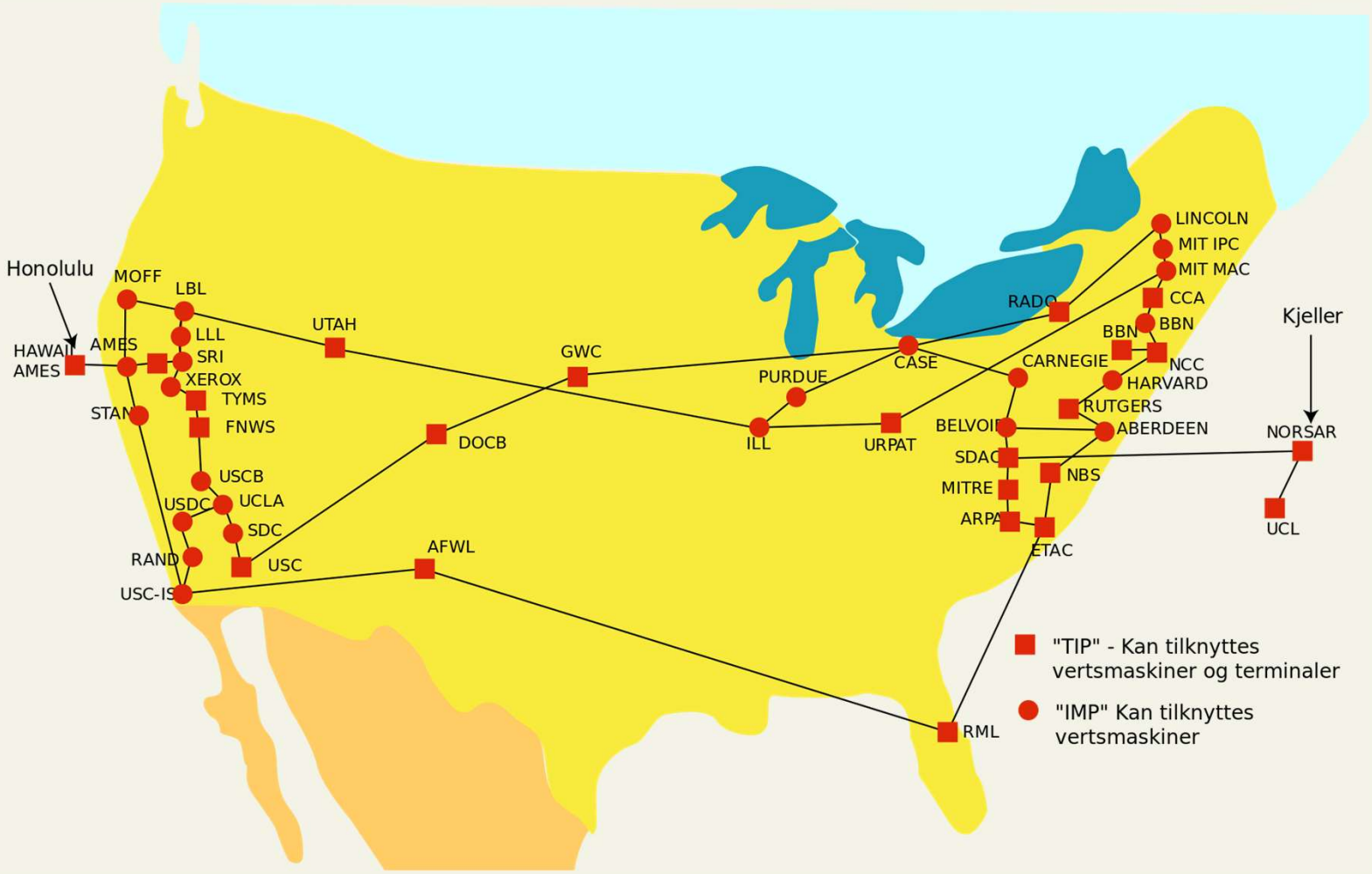
A Short History of the Internet

- Telephone Network
- Packet Networks
  - ARPANET (1969)
  - X.25 (1974)
  - USENET (1979)
  - TCP/IP (1983) ← INTERNET

Definitions and History

A Short History of the Internet

- Map of the ARPANET network in 1974

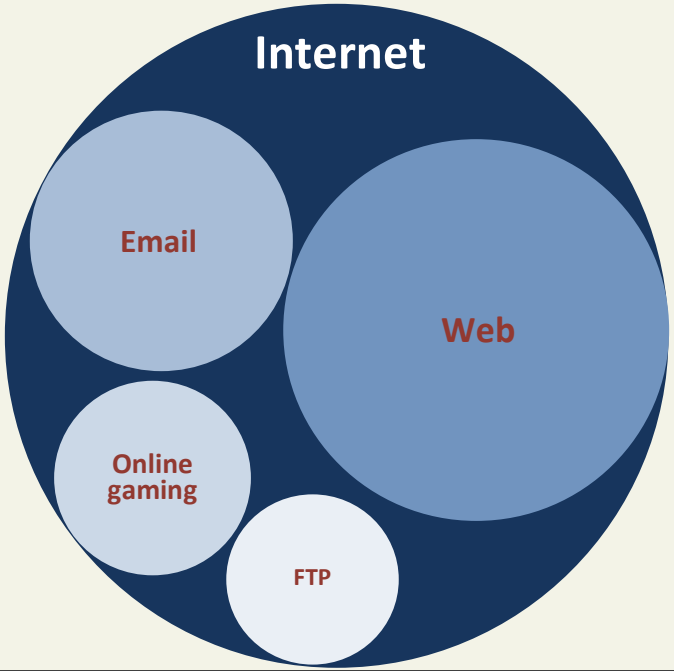


Internet = Web?

The answer is no

The World-Wide Web (WWW or simply the Web) is certainly what most people think of when they see the word “internet.”

But the WWW is only a subset of the Internet.



Communication Definitions

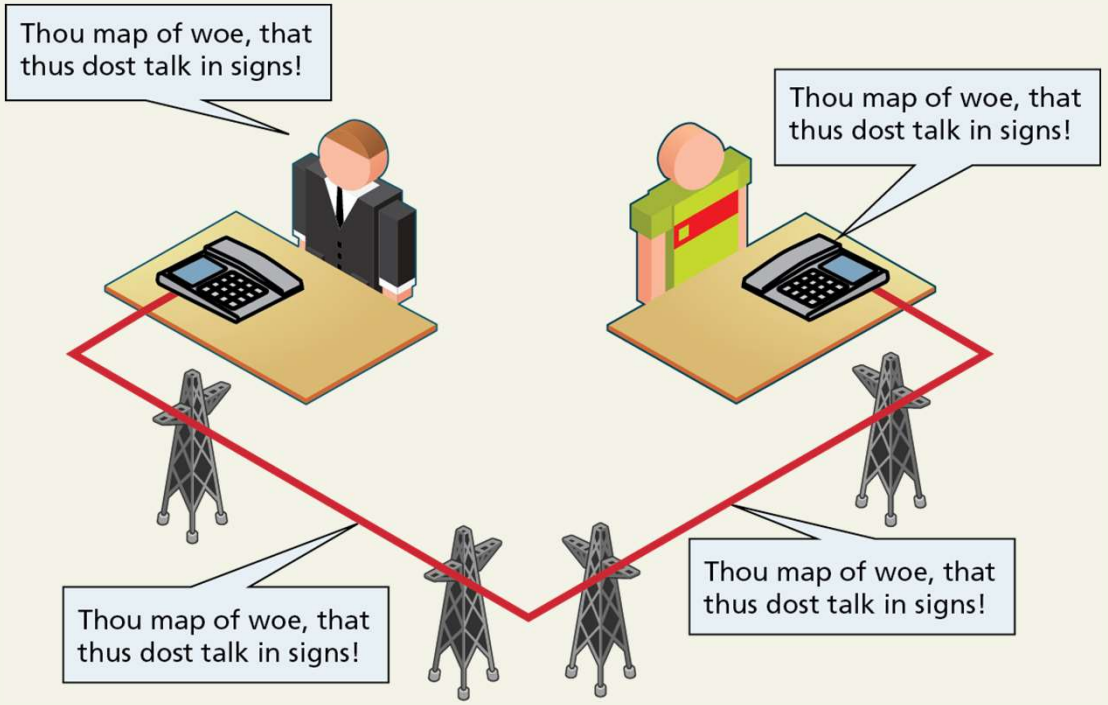
We will begin with the telephone

Telephone networks provide a good starting place to learn about modern digital communications.

In the telephone networks of old, calls were routed through operators who physically connected caller and receiver by connecting a wire to a switchboard to complete the circuit.

Circuit Switching

A **circuit switching** establishes an actual physical connection between two people through a series of physical switches.



Circuit Switching

Its Limitations

Circuit Switching Weaknesses

- You must establish a link and maintain a dedicated circuit for the duration of the call
- Difficult to have multiple conversations simultaneously
- Wastes bandwidth since even the silences are transmitted

ARPANET

The beginnings of the Internet

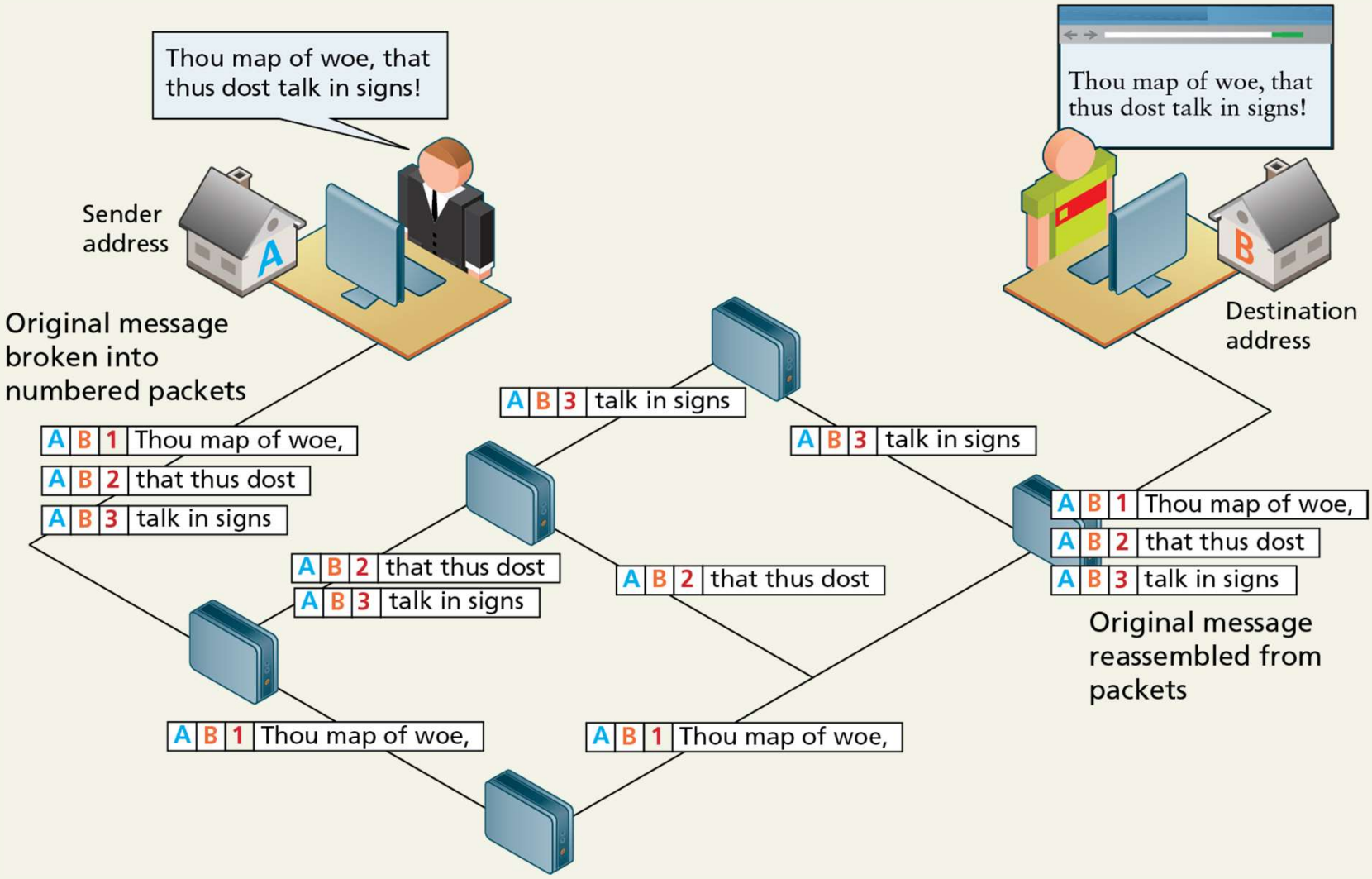
The research network ARPANET was created. In the 1960s

- ARPANET did not use circuit switching
- Used **packet switching**

A packet-switched network does not require a continuous connection. Instead it splits the messages into smaller chunks called **packets** and routes them to the appropriate place based on the destination address.

The **packets** can take different routes to the destination.

Packet Switching



Packet Switching

Isn't this more complicated?

While **packet switching** may seem a more complicated and inefficient approach than **circuit switching**, it is:

- More robust (it is not reliant on a single pathway that may fail) and
- More efficient use of network resources (since a circuit can communicate multiple connections).

Short History of the Internet

Perhaps not short enough

The early ARPANET network was funded and controlled by the United States government, and was used exclusively for academic and scientific purposes.

The early network started small with just a handful of connected campuses in 1969 and grew to a few hundred by the early 1980s.

TCP/IP

Rides to the rescue

To promote the growth and unification of the disparate networks a suite of protocols was invented to unify the networks together.

By 1981, new networks built in the US began to adopt the **TCP/IP** (**Transmission Control Protocol / Internet Protocol**) communication model, while older networks were transitioned over to it.

Tim Berners-Lee

I meant Sir Tim Berners-Lee

The invention of the WWW is usually attributed to the British Tim Berners-Lee, who, along with the Belgian Robert Cailliau, published a proposal in 1990 for a hypertext system while both were working at CERN in Switzerland.

Core Features of the Web

Shortly after that initial proposal Berners-Lee developed the main features of the web:

1. A URL to uniquely identify a resource on the WWW.
2. The HTTP protocol to describe how requests and responses operate.
3. A software program (later called web server software) that can respond to HTTP requests.
4. HTML to publish documents.
5. A program (later called a browser) to make HTTP requests from URLs and that can display the HTML it receives.

W3C

The World Wide Web Consortium

Also in late 1994, Berners-Lee helped found the **World Wide Web Consortium (W3C)**, which would soon become the international standards organization that would oversee the growth of the web.

This growth was very much facilitated by the decision of CERN to not patent the work and ideas done by its employee and instead left the web protocols and code-base royalty free.

Static Web Sites

Partying Like It's 1995

In the earliest days of the web, a **webmaster** would publish web pages, and periodically update them.

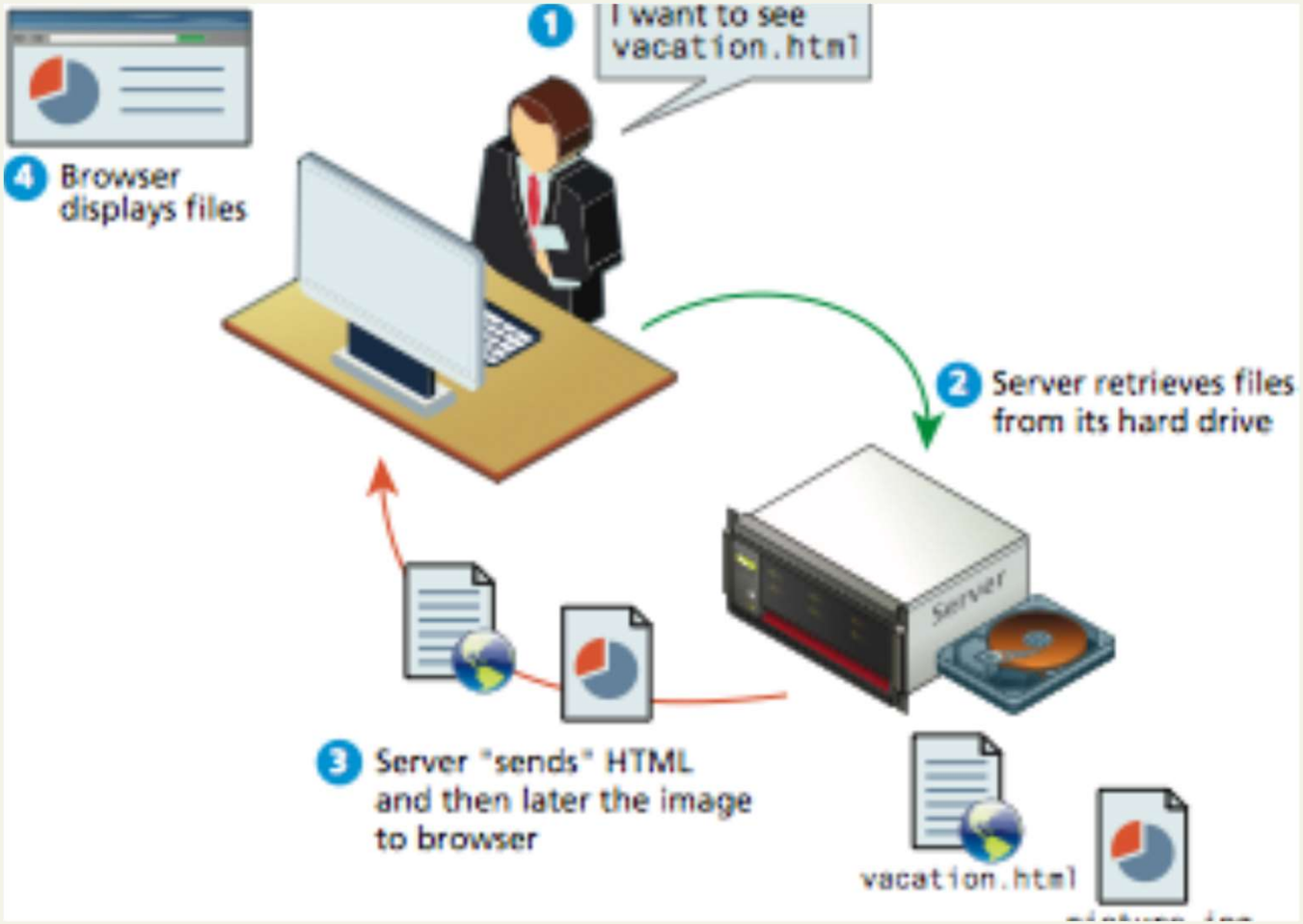
In those early days, the skills needed to create a web site were pretty basic:

- Knowledge of the HTML markup language and perhaps familiarity with editing and creating images

Commonly referred to as a **static web site**, in that it consists only of HTML pages that look identical for all users at all times.

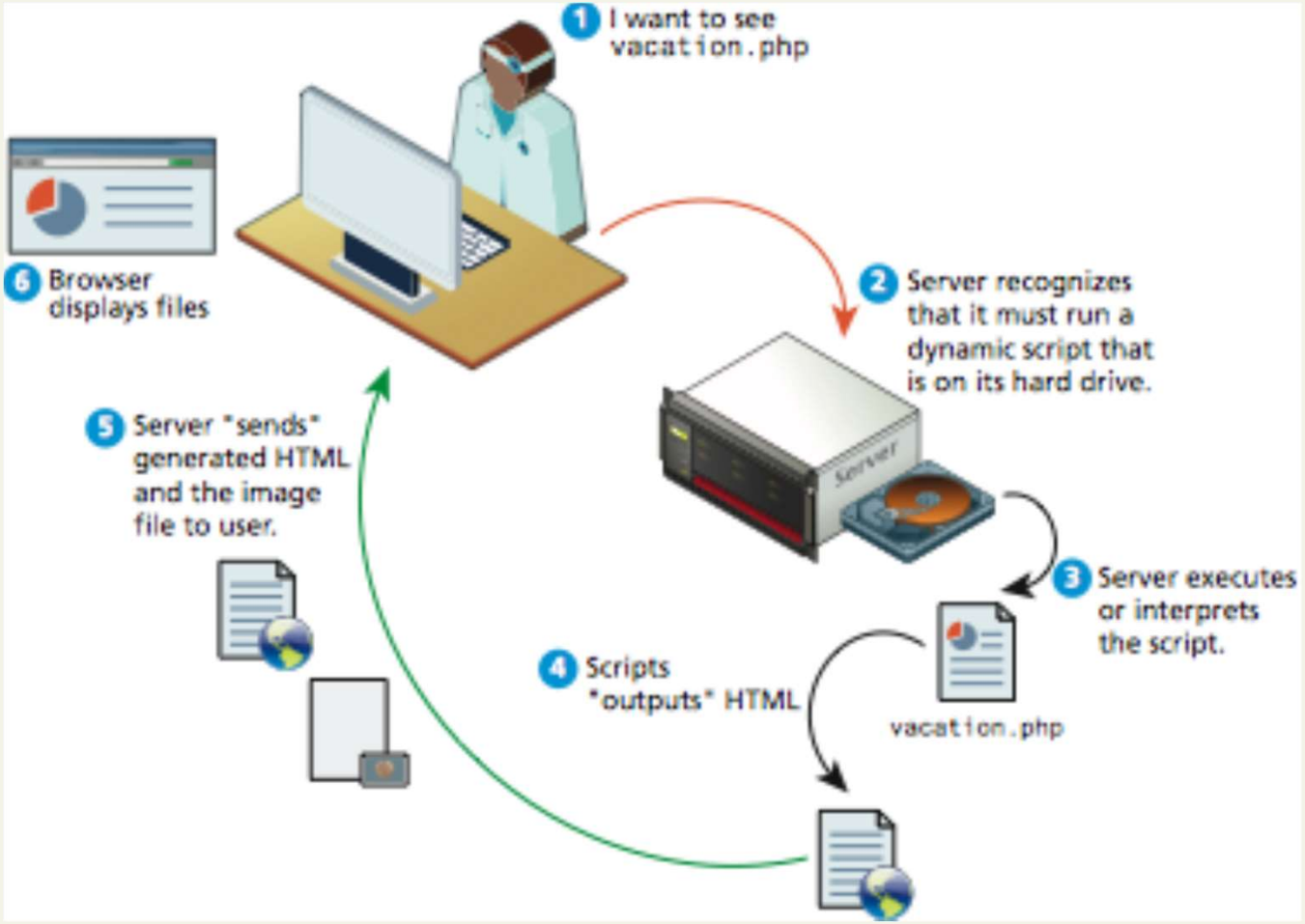
Definitions and History

Static Websites versus Dynamic Websites



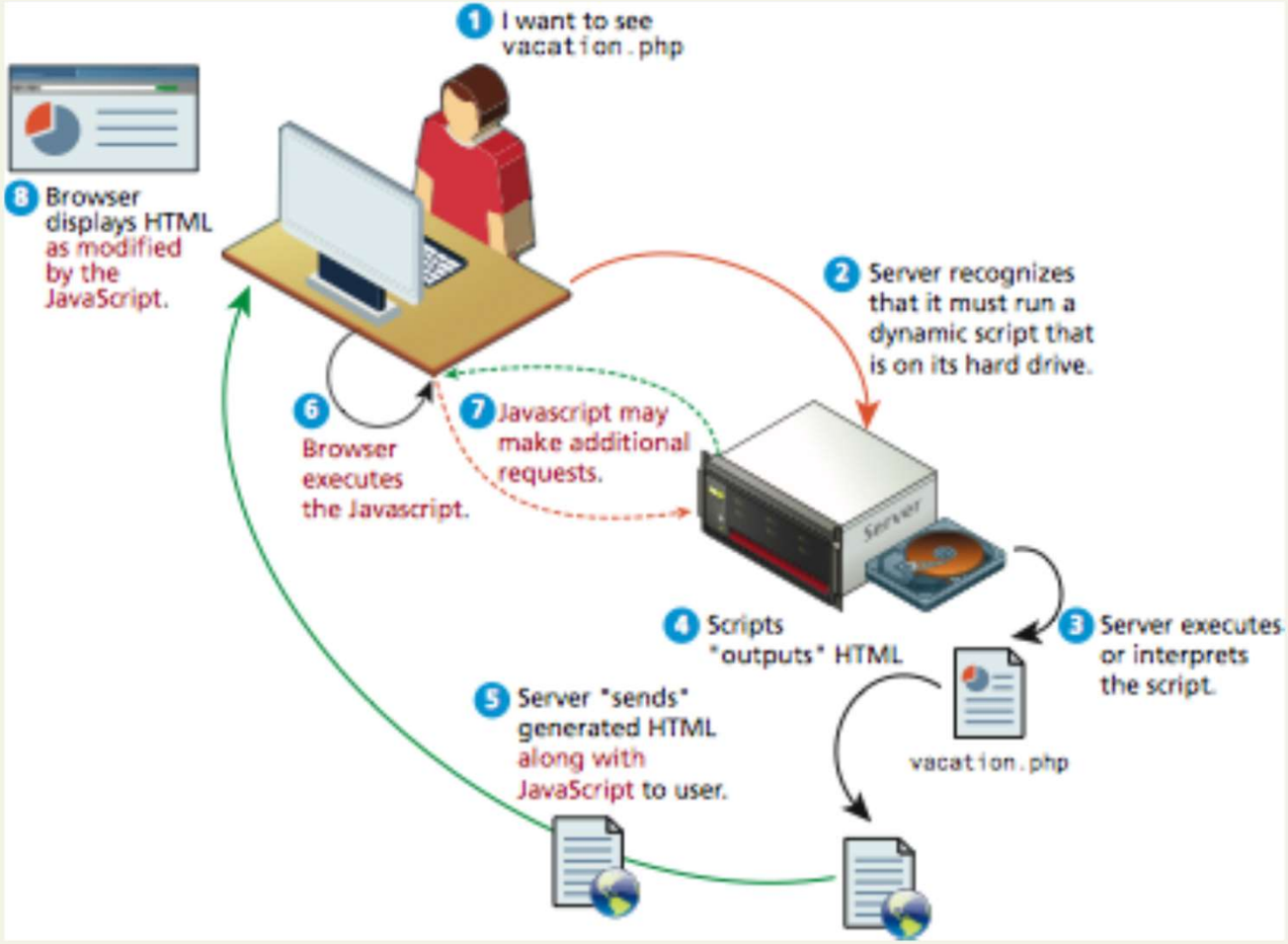
Definitions and History

Static Websites versus Dynamic Websites



Definitions and History

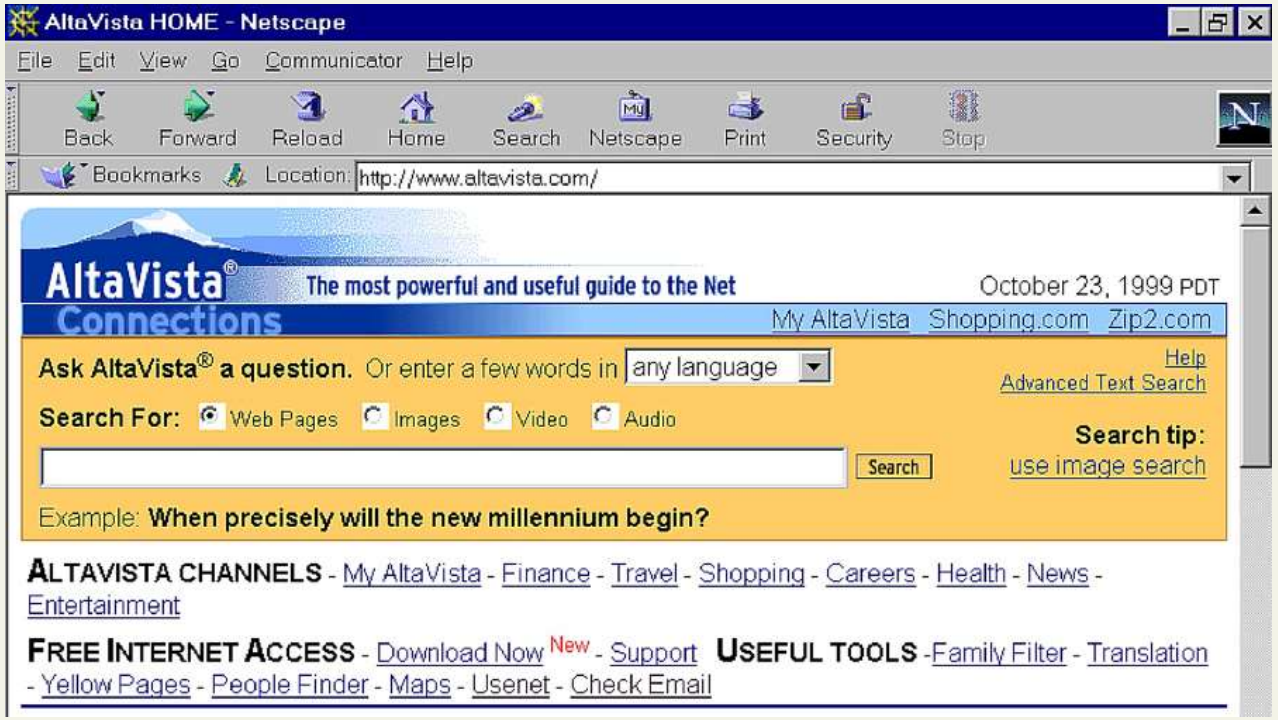
Static Websites versus Dynamic Websites



Web 2.0 and Beyond

In the mid 2000s, a new buzz-word entered the computer lexicon: **web 2.0**.

This term had two meanings, one for users and one for developers.



Web 2.0

Its meaning for users

For the users, Web 2.0 referred to an interactive experience where users could contribute *and* consume web content, thus creating a more user-driven web experience.

Web 2.0

Its meaning for developers

For software developers, Web 2.0 also referred to a change in the paradigm of how dynamic web sites are created.

Programming logic, which previously existed only on the server, began to migrate to the browser.

This required learning Javascript, a rather tricky programming language that runs in the browser, as well as mastering the rather difficult programming techniques involved in asynchronous communication.

Chapter 1

1. A Complicated Ecosystem
2. Definitions and History
3. The Client-Server Model
4. Where is the Internet?
5. Summary

Client-Server Model

What is it?

The web is sometimes referred to as a client-server model of communications.

In the **client-server model**, there are two types of actors: clients and servers.

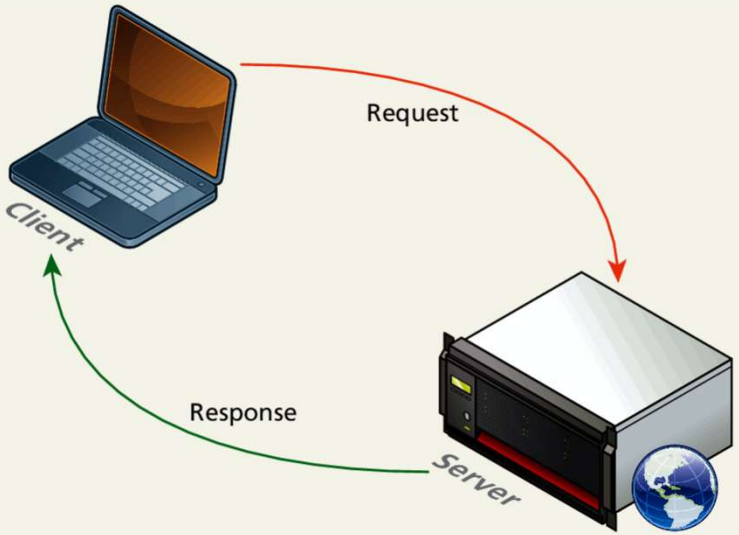
The **server** is a computer agent that is normally active 24 hours a day, 7 days a week (or simply 24/7), listening for queries from any client who make a request.

A **client** is a computer agent that makes requests and receives responses from the server, in the form of response codes, images, text files, and other data.

Request-Response Loop

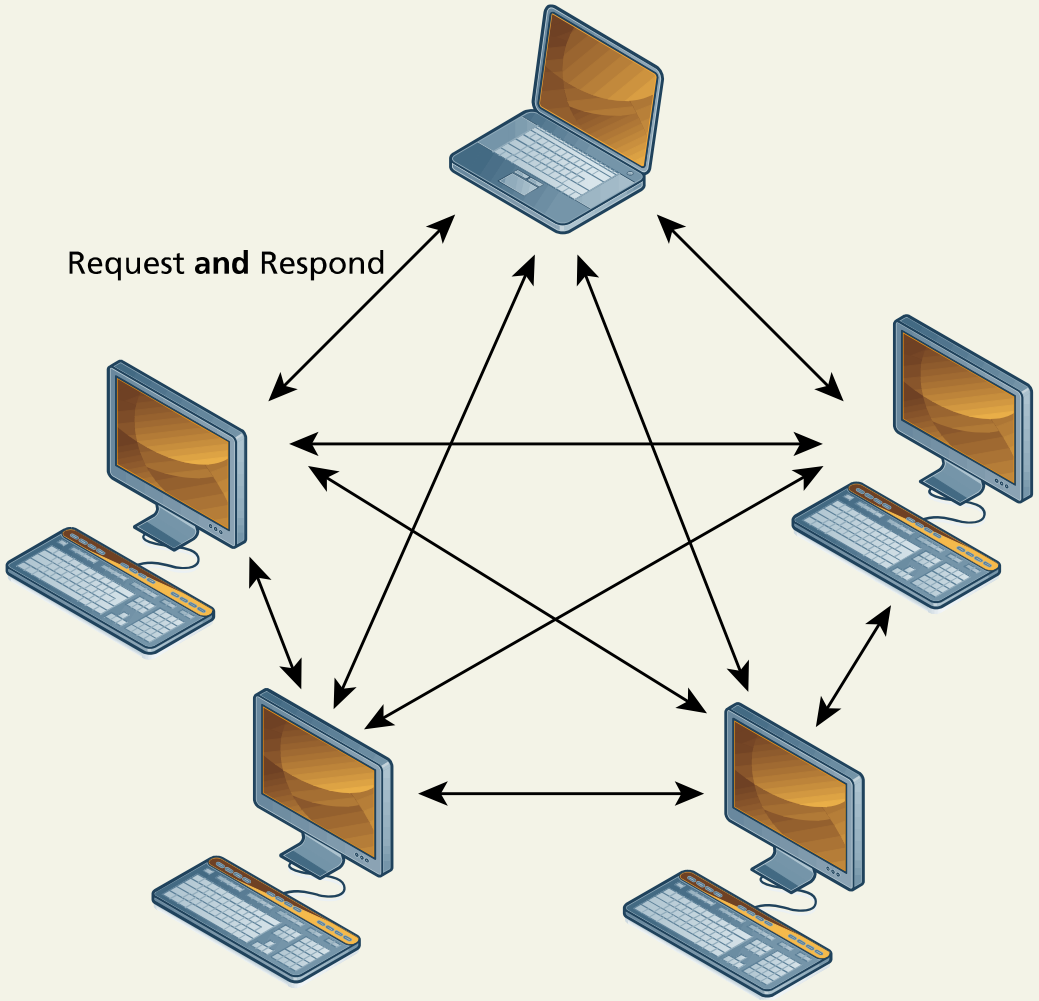
Within the client-server model, the **request-response loop** is the most basic mechanism on the server for receiving requests and transmitting data in response.

The client initiates a **request** to a server and gets a **response** that could include some resource like an HTML file, an image or some other data.



The Client-Server Model

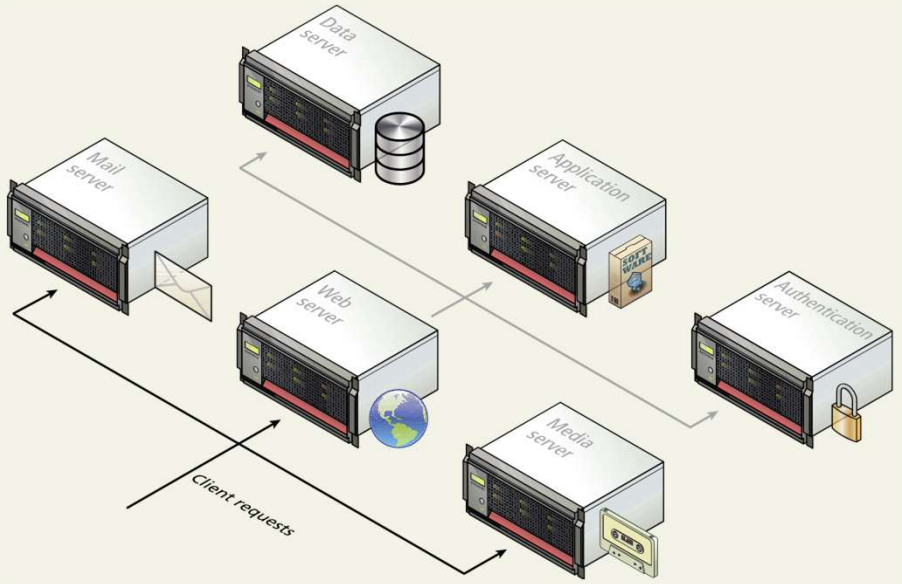
The Peer-to-Peer Alternative



The Client-Server Model

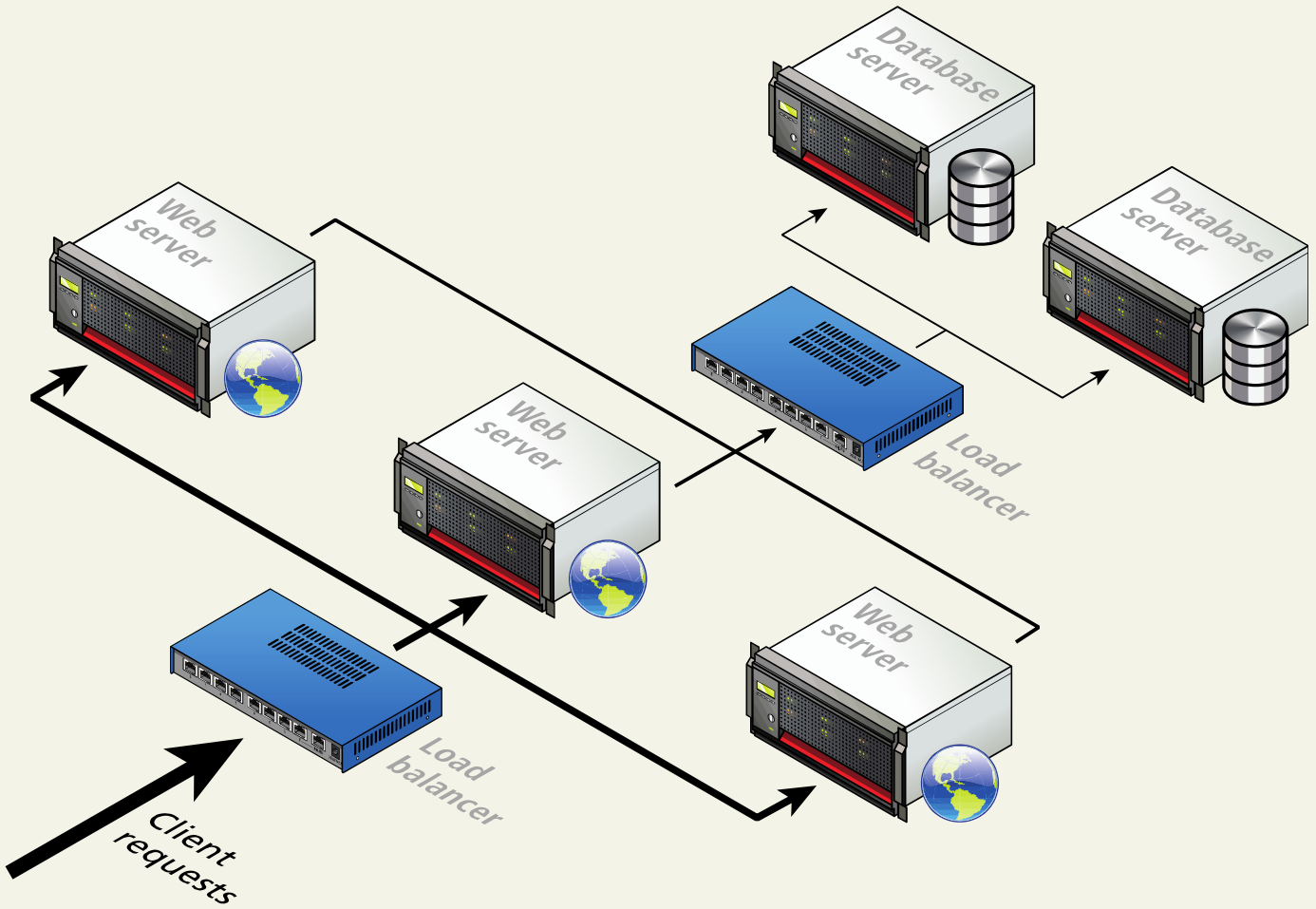
Server Types

- Web Servers
- Application Servers
- Database Servers
- Mail Servers
- Media Servers
- Authentication Servers



The Client-Server Model

Real-World Server Installations – Server Farm

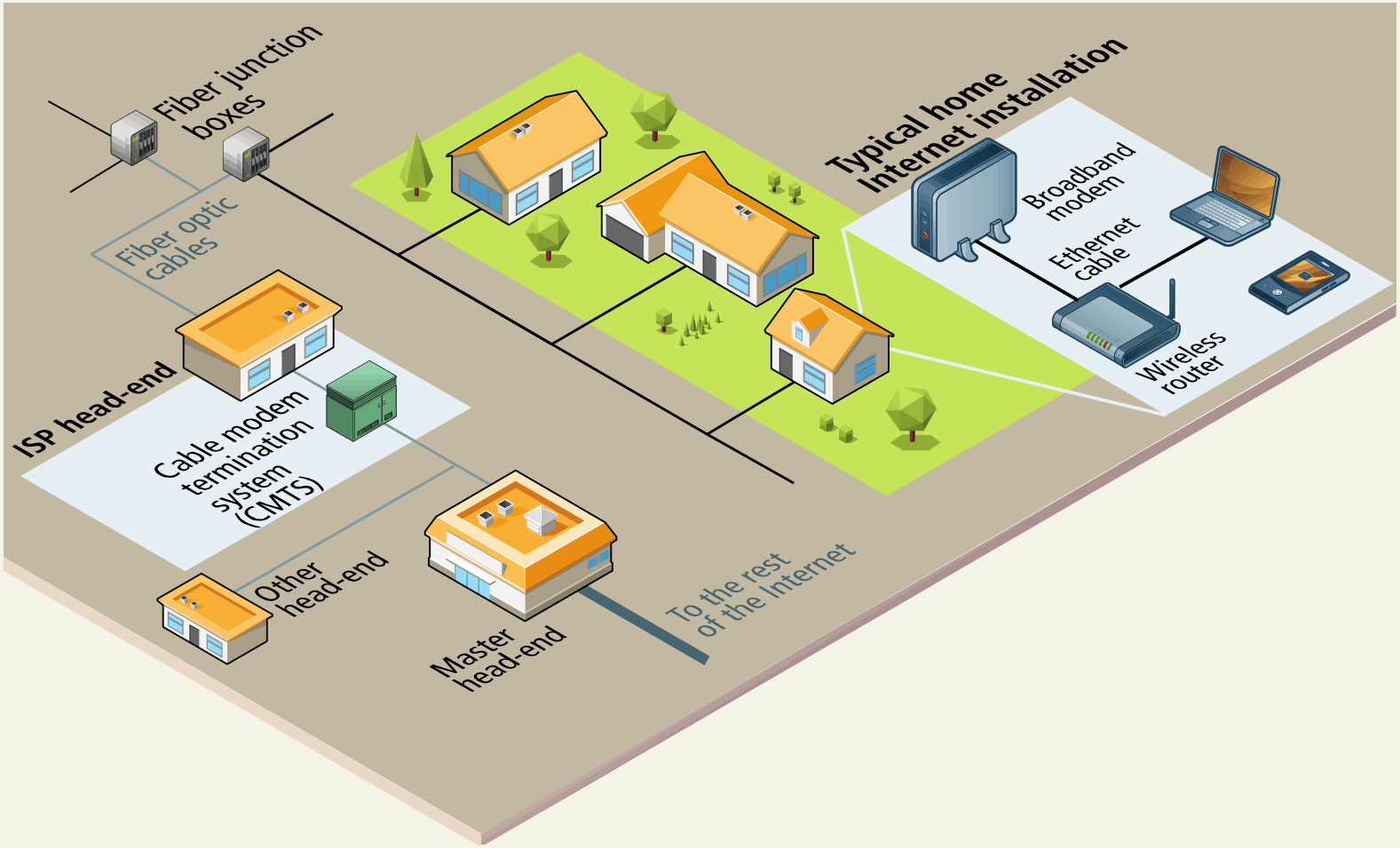


Chapter 1

1. A Complicated Ecosystem
2. Definitions and History
3. The Client-Server Model
4. Where is the Internet?
5. Summary

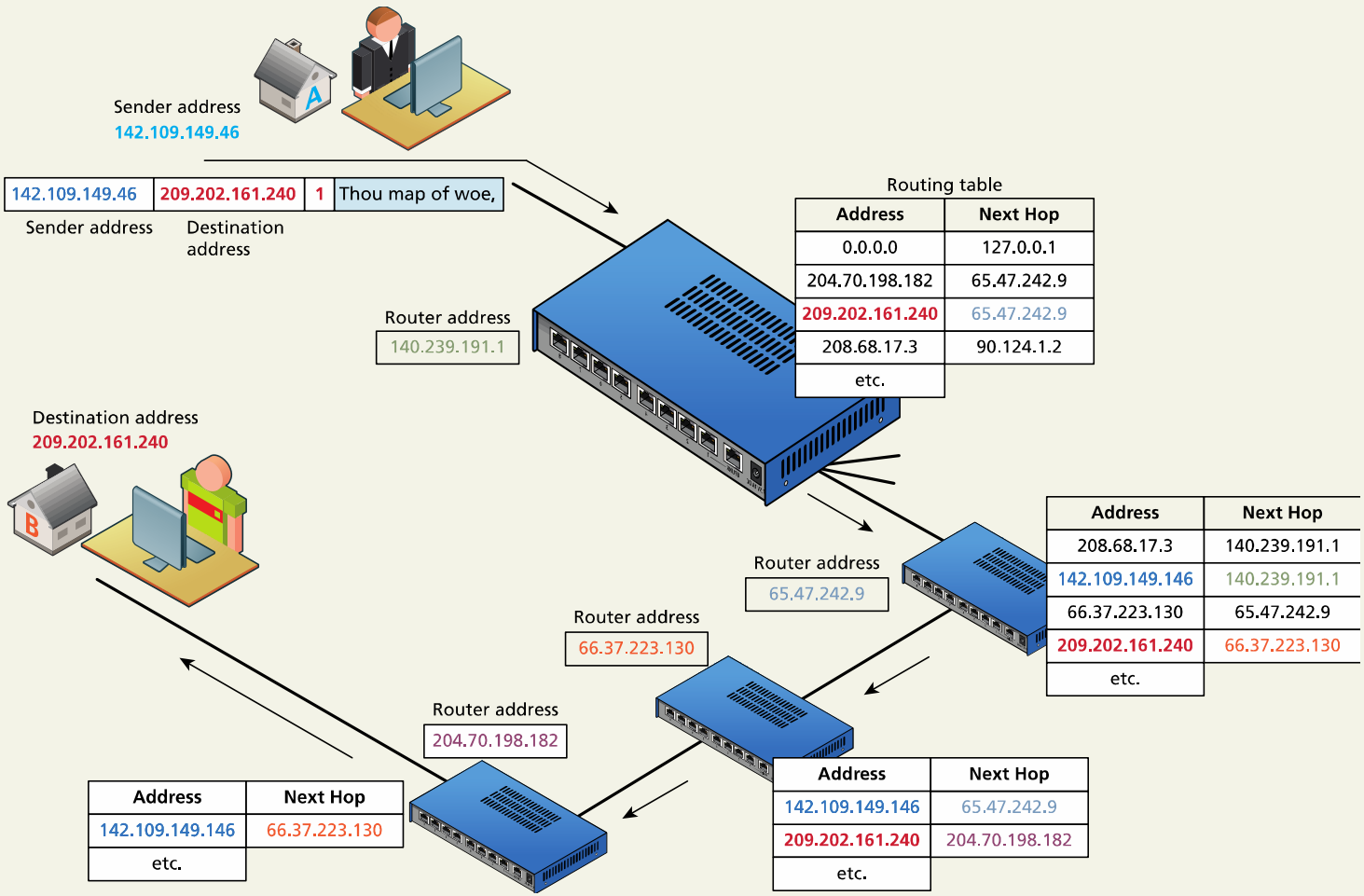
Where Is the Internet?

From the Computer to the Local Provider



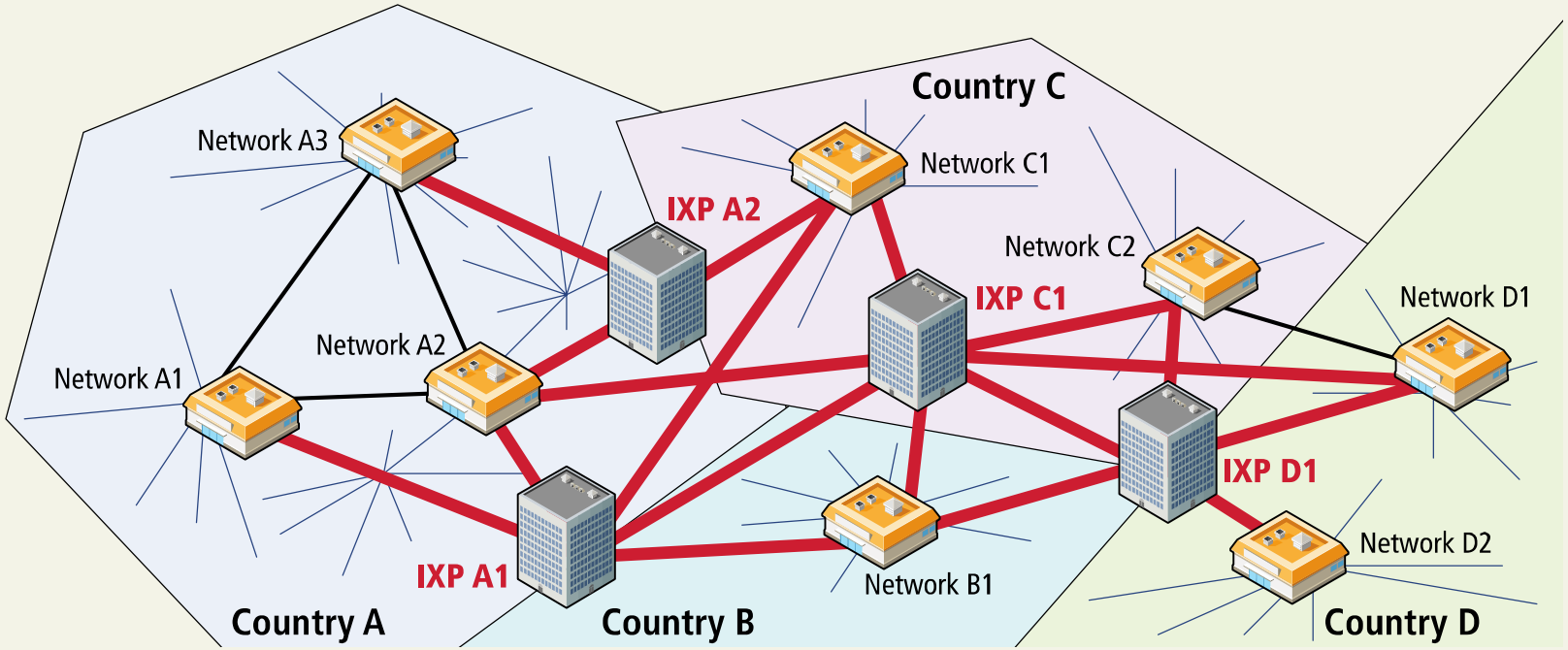
Where Is the Internet?

(Simplified) Routing Tables



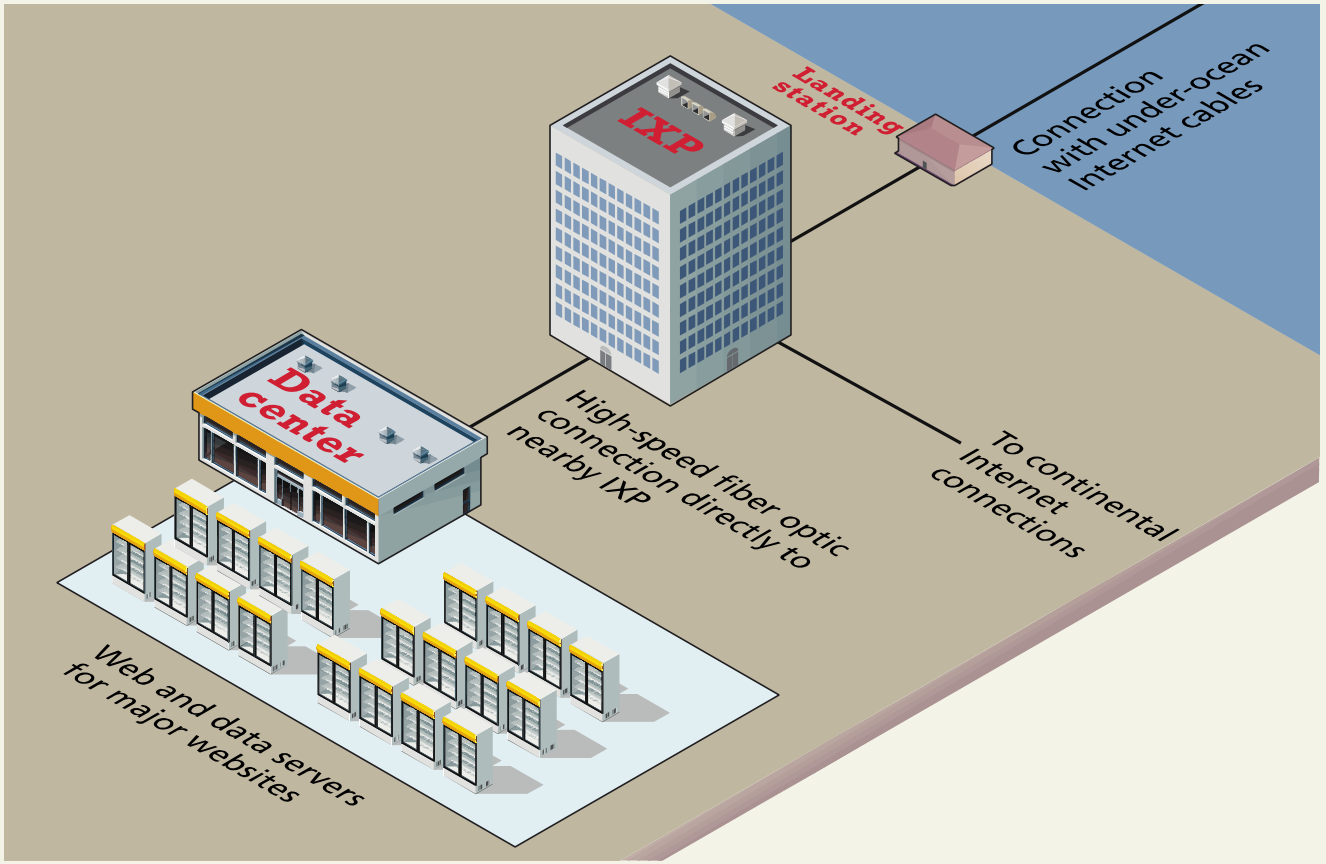
Where Is the Internet?

From the Local Provider to the Ocean's Edge



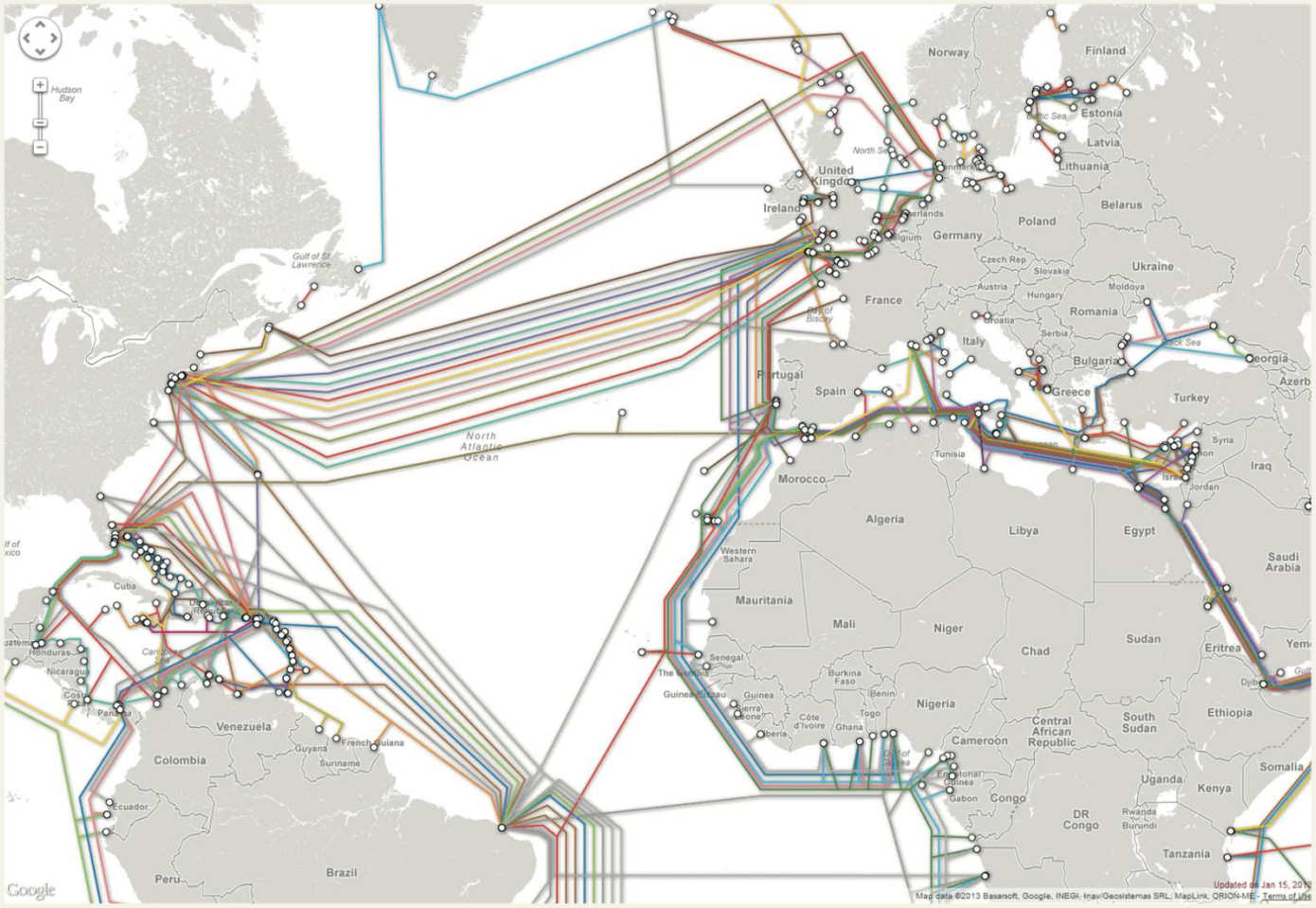
Where Is the Internet?

From the Local Provider to the Ocean's Edge –IXP and Data Centers



Where Is the Internet?

Across the Oceans



Chapter 2

1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

What's a Protocol?

The internet exists today because of a suite of interrelated communications protocols.

A **protocol** is a set of rules that partners in communication use when they communicate.

TCP/IP protocols have been implemented in every operating system, and make fast web development possible.

Networking is it's own entire discipline.

Web developer needs general awareness of what the suite of Internet protocols does

A Layered Architecture

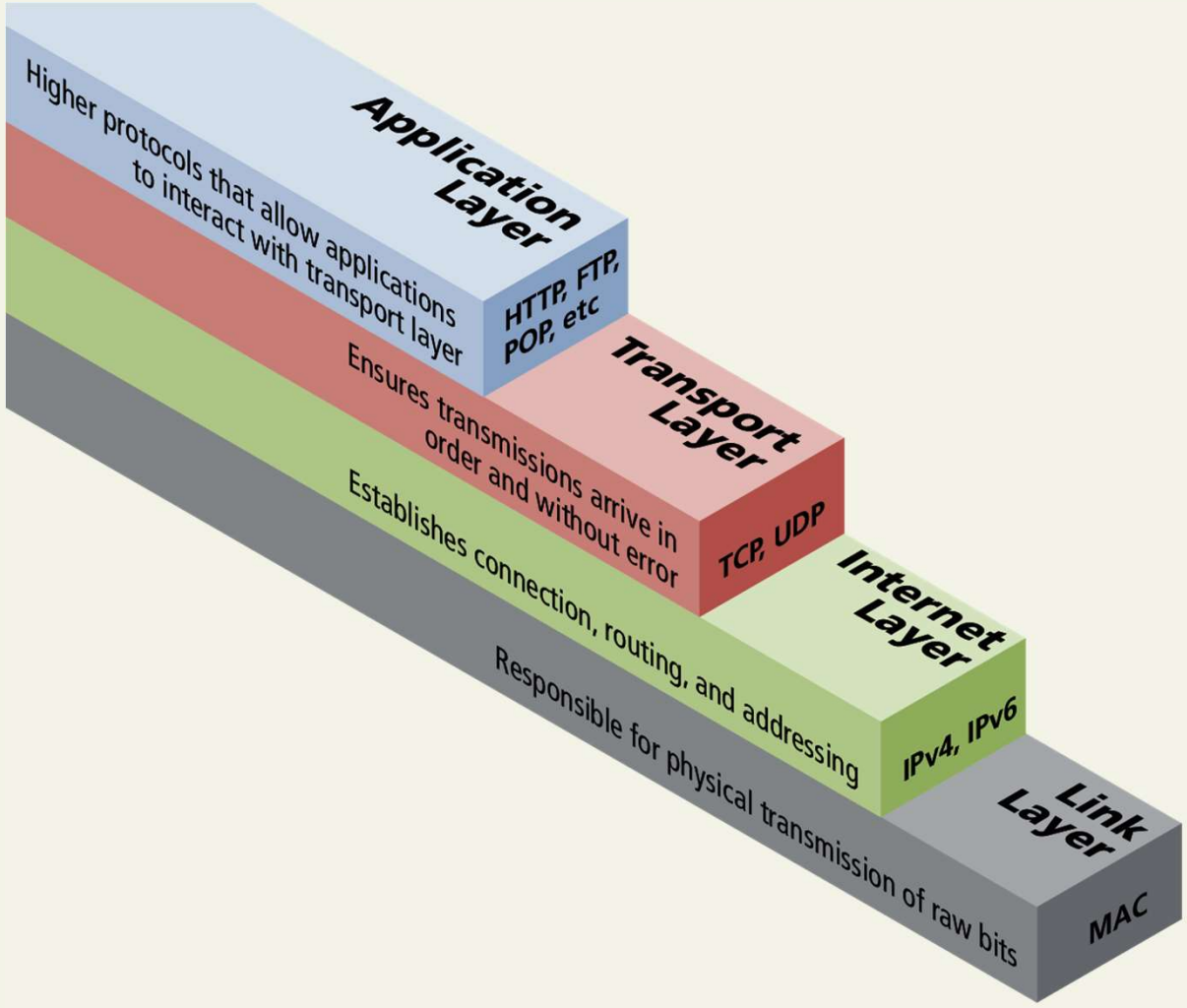
The TCP/IP Internet protocols were originally abstracted as a four-layer stack.

Later abstractions subdivide it further into five or seven layers.

Focused on the top layer

- Use the earliest and simplest **four-layer network model** (actual model is 5 layers)

Four Layer Network Model



Link Layer

L2

The **link layer** is the lowest layer, responsible

- physical transmission of data across media (both wired and wireless) and
- Establishing logical links.

It handles issues like transmission, collisions, line sharing and more.

Internet Layer

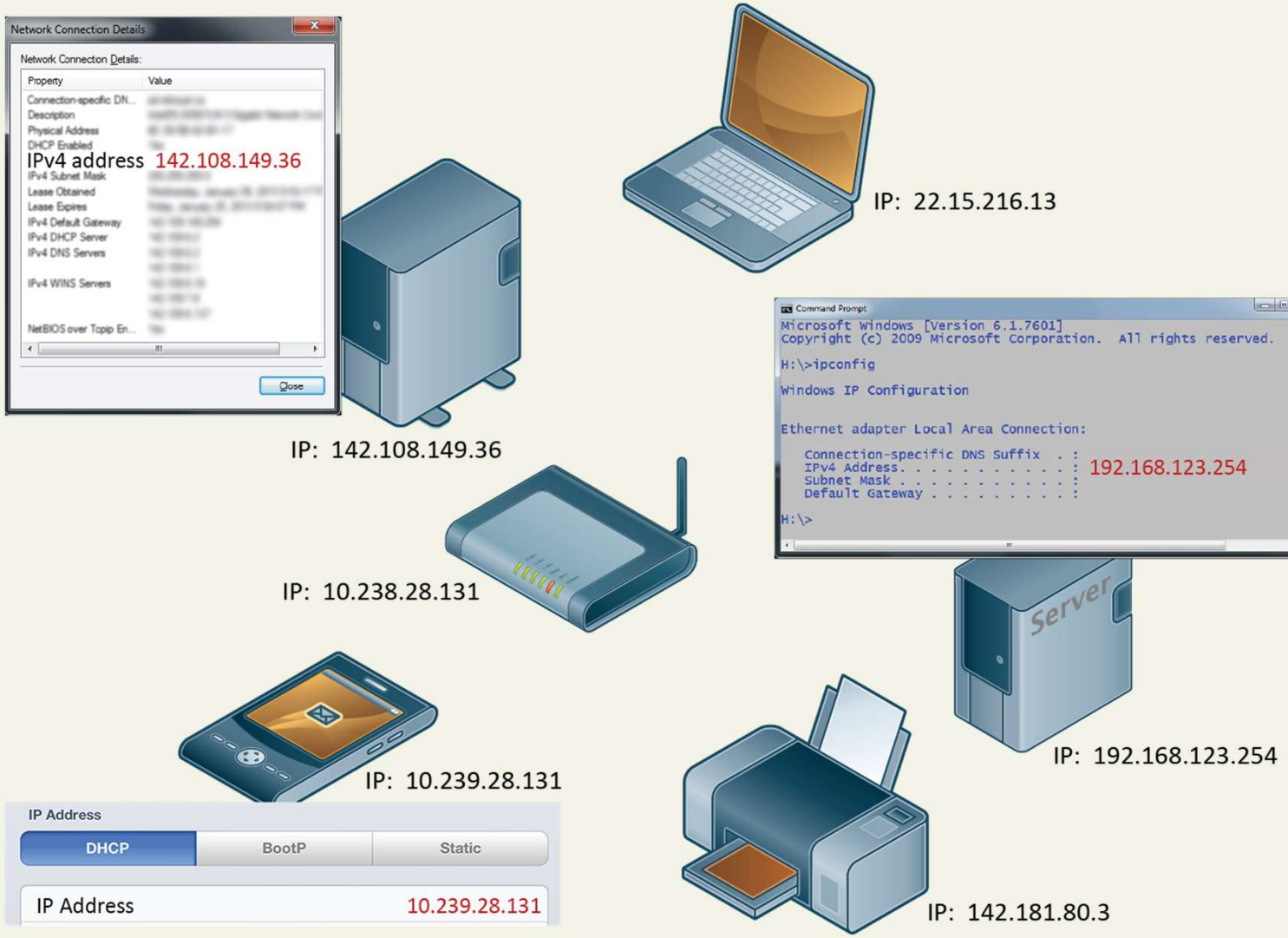
L3

The **internet layer** (sometimes also called the IP Layer) routes packets between communication partners across networks.

The Internet uses the **Internet Protocol** (**IP**) addresses to identify destinations on the Internet.

Every device connected to the Internet has an **IP address**, which is a numeric code that is meant to uniquely identify it.

IP addresses and the Internet



IP Addresses

Two types

IPv4 addresses are the IP addresses from the original TCP/IP protocol.

In IPv4, 12 numbers are used (implemented as four 8-bit integers), written with a dot between each integer.

Since an unsigned 8-bit integer's maximum value is 255, four integers together can encode approximately 4.2 billion unique IP addresses.

IP Addresses

Two types

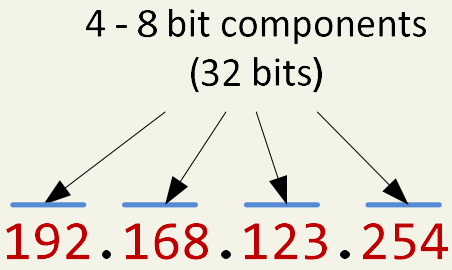
To future proof the Internet against the 4.2 billion limit, a new version of the IP protocol was created, **IPv6**.

This newer version uses eight 16-bit integers for 2^{128} unique addresses, over a billion billion times the number in IPv4.

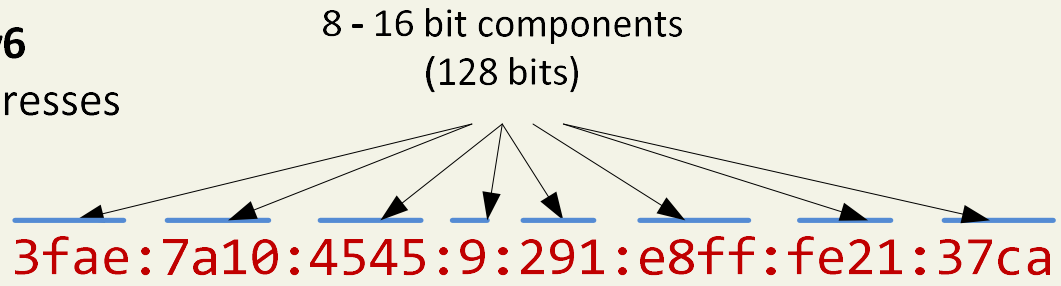
These 16-bit integers are normally written in hexadecimal, due to their longer length.

IPv4 V. IPv6

IPv4
2^{32} addresses



IPv6
2^{128} addresses



Transport Layer

L4

The **transport layer** ensures transmissions arrive, in order, and without error.

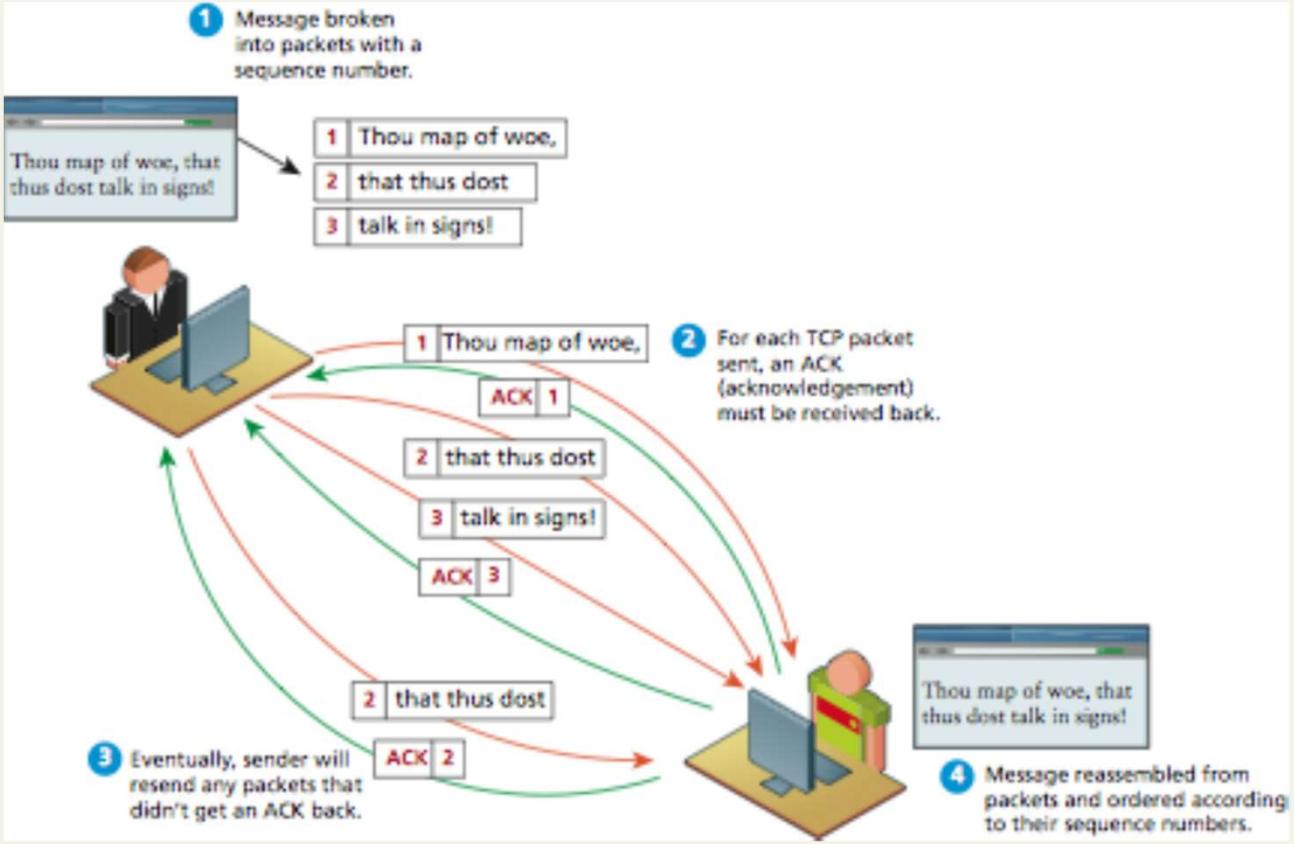
This is accomplished through a few mechanisms.

- Data is broken into packets formatting according to the **Transmission Control Protocol** (**TCP**).
- Each packet is acknowledged back to the sender so in the event of a lost packet, the transmitter will realize a packet has been lost since no ACK arrived for that packet.
    - The lost packet is retransmitted, and although out of order, is reordered at the destination.

Internet Protocols

Transport Layer (TCP)

- Ensures transmissions arrive in order and without error



Application Layer

L5 and Above

There are **many application layer** protocols. Web developers should be aware of :

- **HTTP.** The Hypertext Transfer Protocol is used for web communication.
- **SSH.** The Secure Shell Protocol allows remote command-line connections to servers.
- **FTP**. The File Transfer Protocol is used for transferring files between computers.
- **POP/IMAP/SMTP**. Email-related protocols for transferring and storing email.
- **DNS**. The Domain Name System protocol used for resolving domain names to IP addresses.

Chapter 2

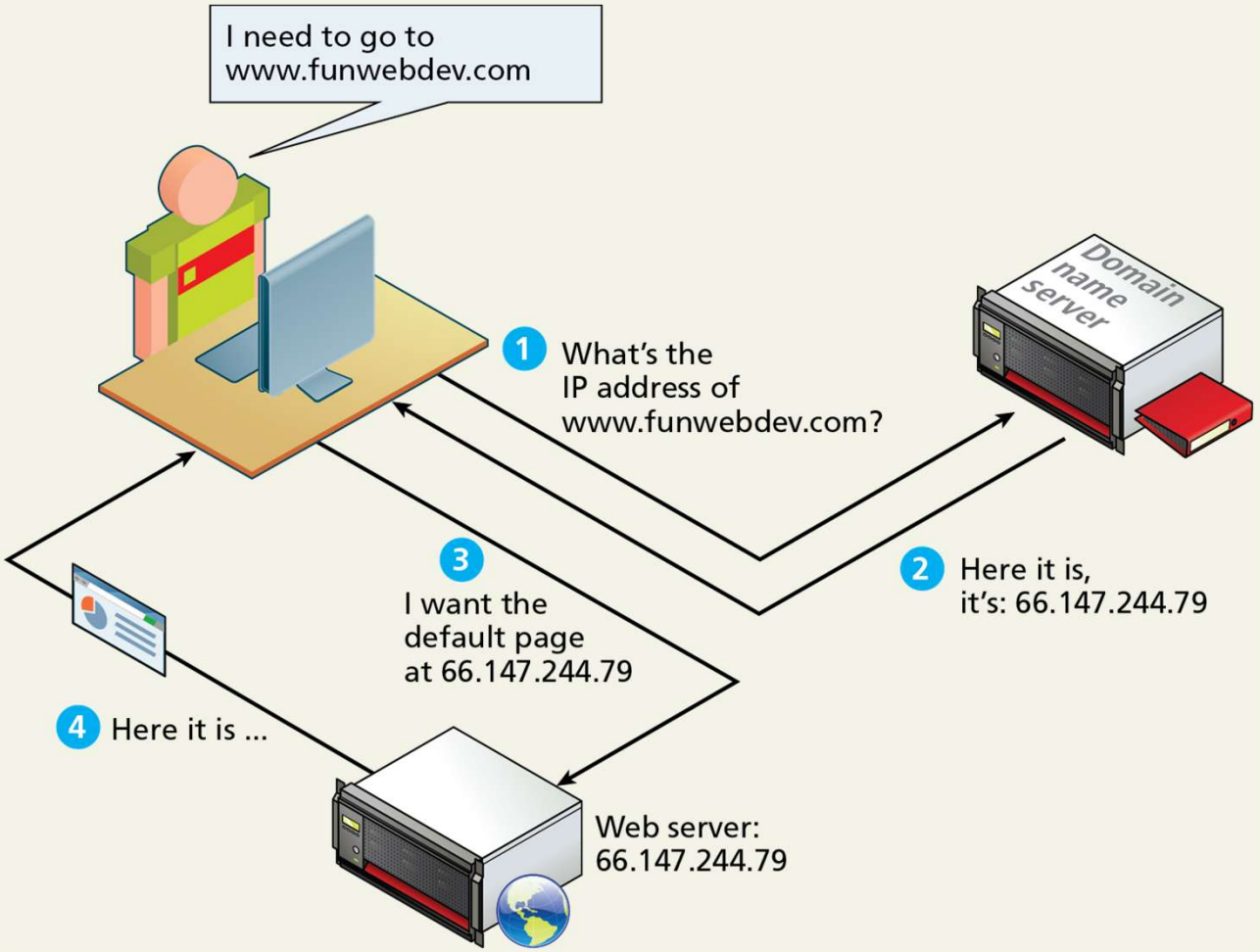
1. Internet Protocols
2. Domain Name System
3. Uniform Resource Locators
4. Hypertext Transfer Protocol
5. Web Browsers
6. Web Servers
7. Summary

Domain Name System

Why do we need it?

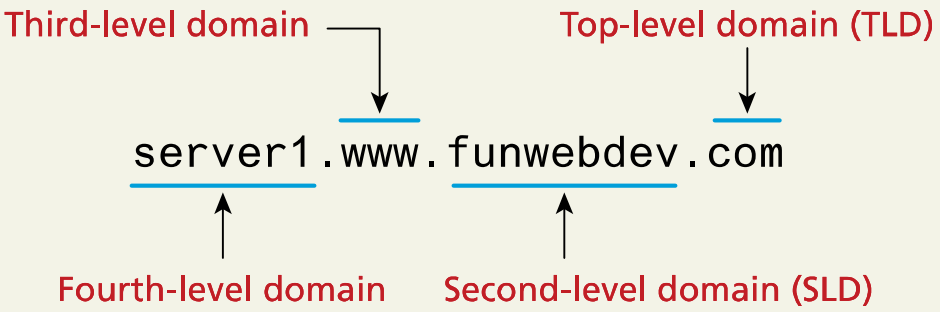
As elegant as IP addresses may be, human beings do not enjoy having to recall long strings of numbers. Instead of IP addresses, we use the **Domain Name System** (**DNS**)

DNS Overview

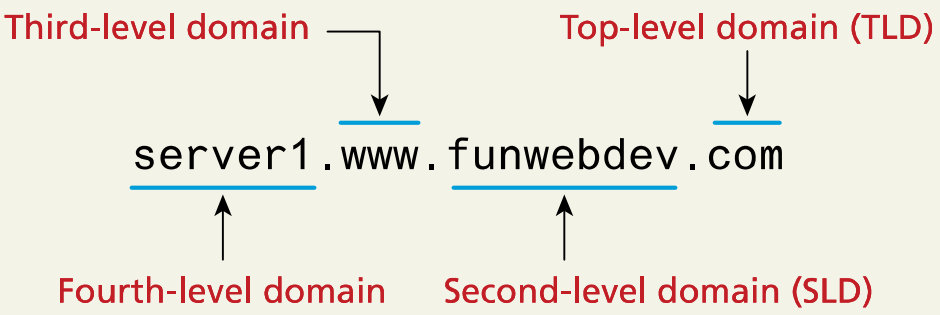


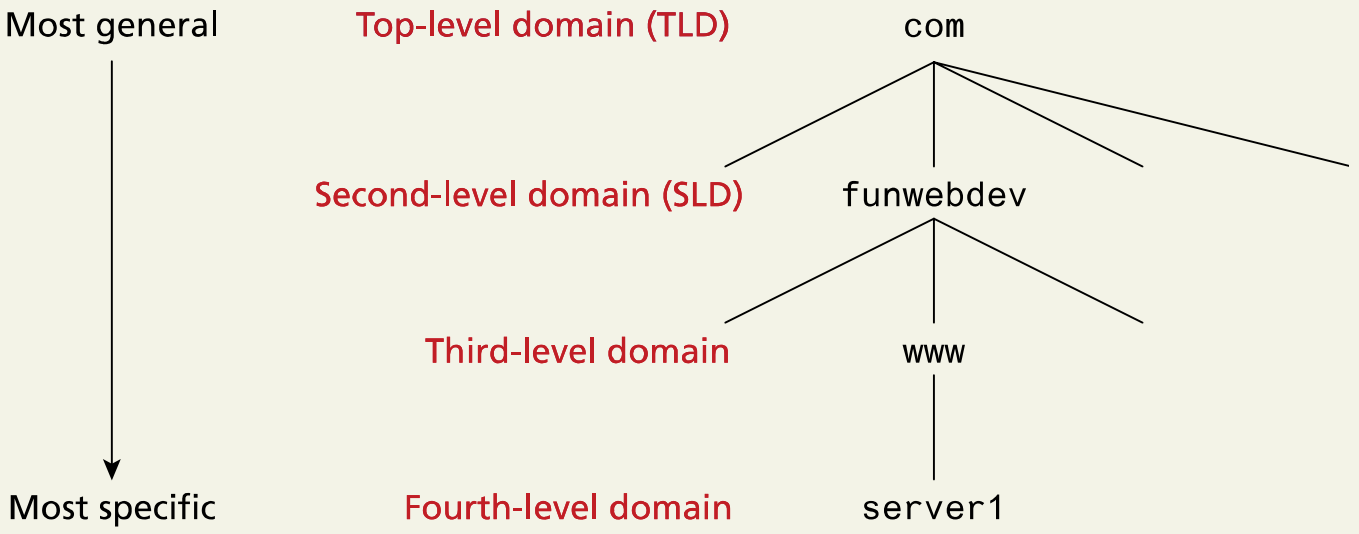
Domain Name System

Name Levels



Domain Levels





Name Registration

How are domain names assigned?

Special organizations or companies called **domain name registrars** manage the registration of domain names.

These domain name registrars are given permission to do so by the appropriate generic top-level domain (gTLD) registry and/or a country code top-level domain (ccTLD) registry.

Types of TLDs

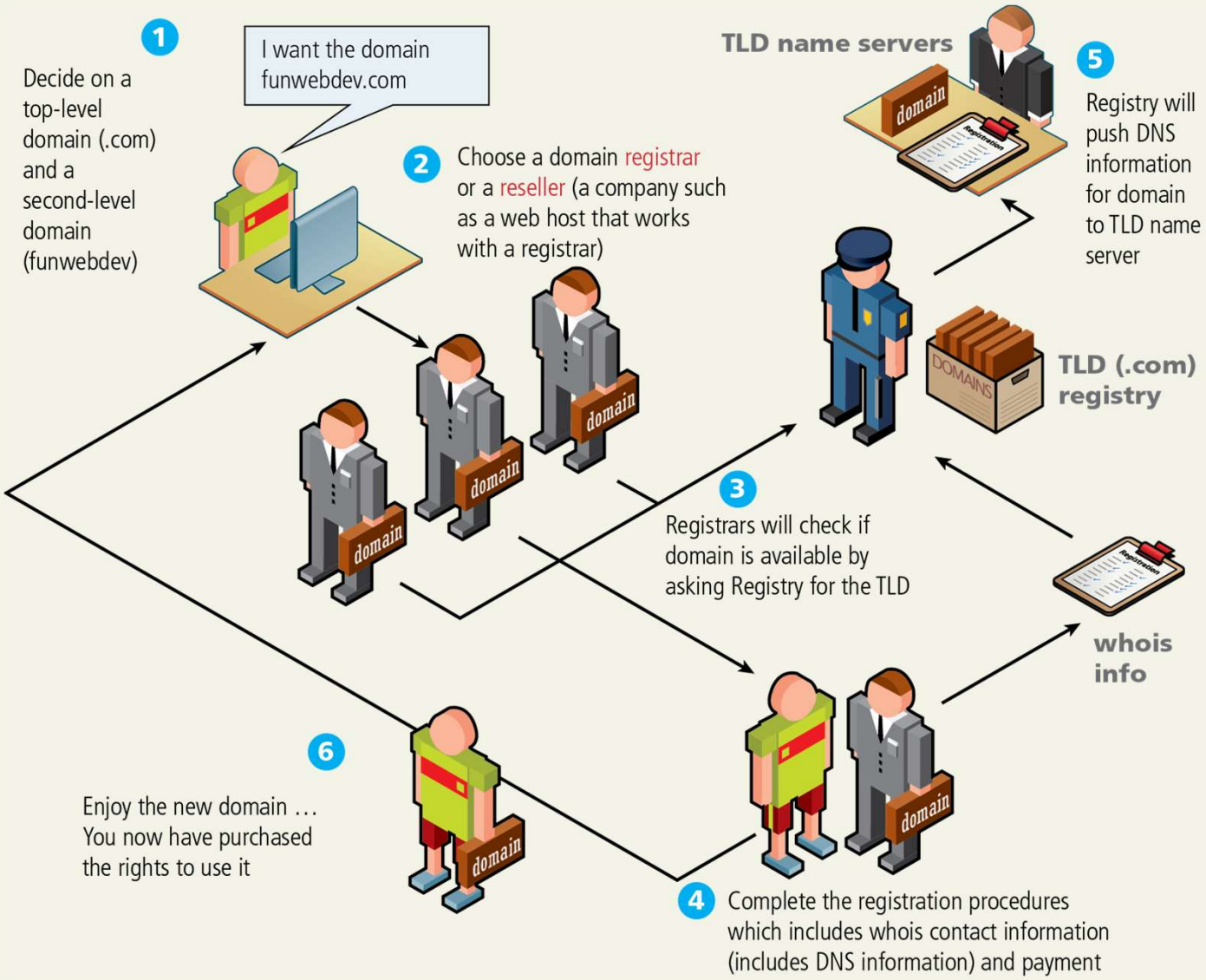
Generic top-level domains (gTLD)

- Unrestricted. TLDs include .com, .net, .org, and .info.
- Sponsored. TLDs including .gov, .mil, .edu, and others.
- New TLDs.

Country code top-level domain (ccTLD)

- TLDs include .us , .ca , .uk , and .au.
- Internationalized Domain Names

Domain name registration process



DNS Address Resolution

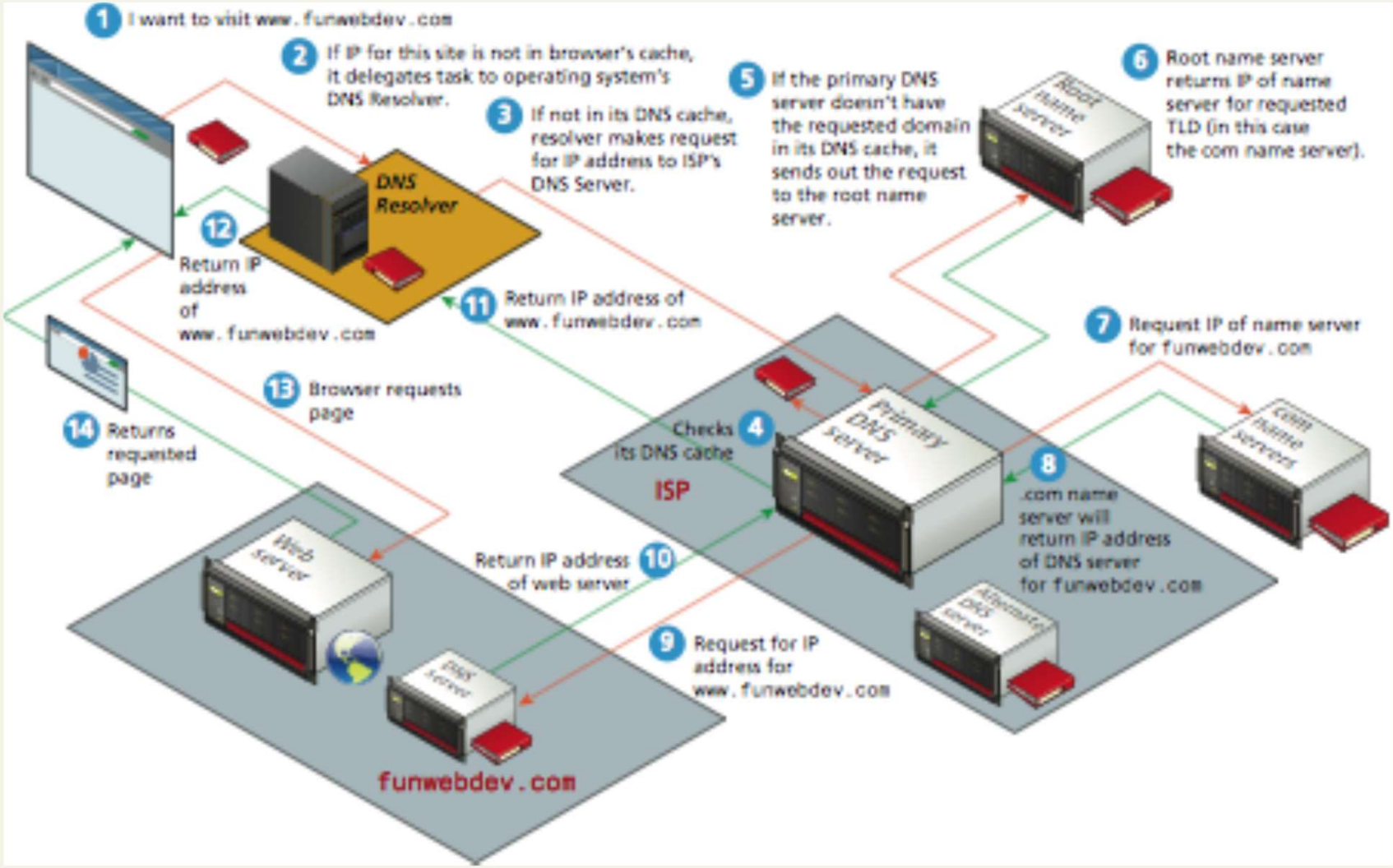
While domain names are certainly an easier way for users to reference a web site, eventually, your browser needs to know the IP address of the web site in order to request any resources from it.

The Domain Name System provides a mechanism for software to discover this numeric IP address.

This process is referred to here as **address resolution**.

Domain Name System

Address Resolution Process



Question

Which of the following organizations oversees the management of top-level domains, accredits registrars, and coordinates other aspects of DNS?

A. W3C

B. ICANN

C. ARPANET

D. IANA

Question

Which of the following organizations oversees the management of top-level domains, accredits registrars, and coordinates other aspects of DNS?

A. W3C

B. ICANN

C. ARPANET

D. IANA