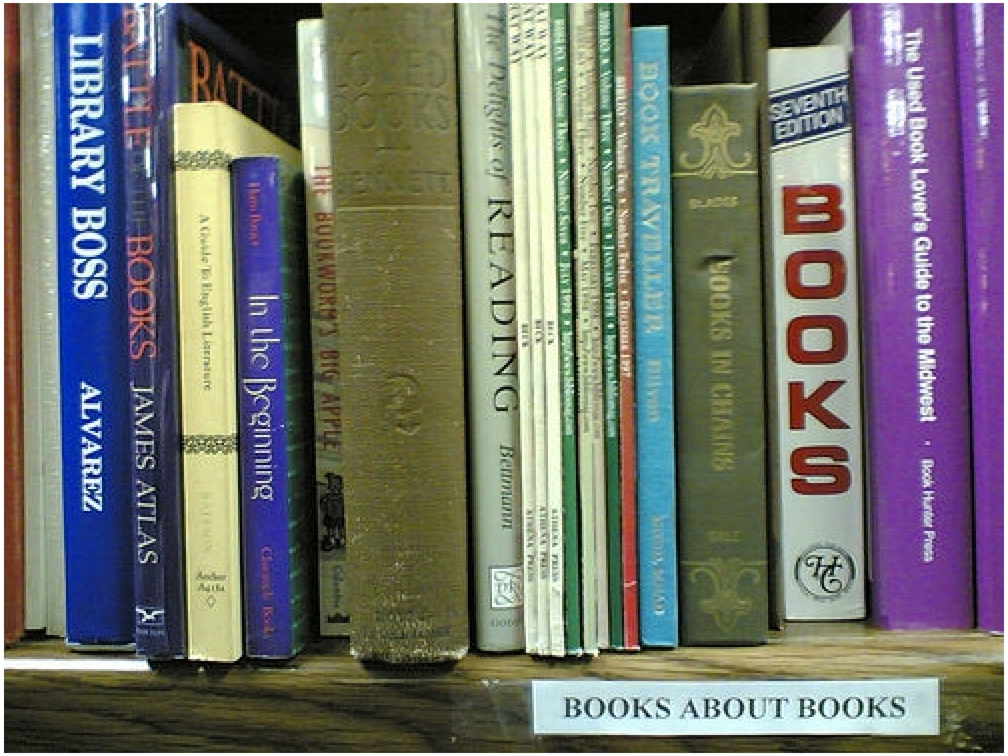
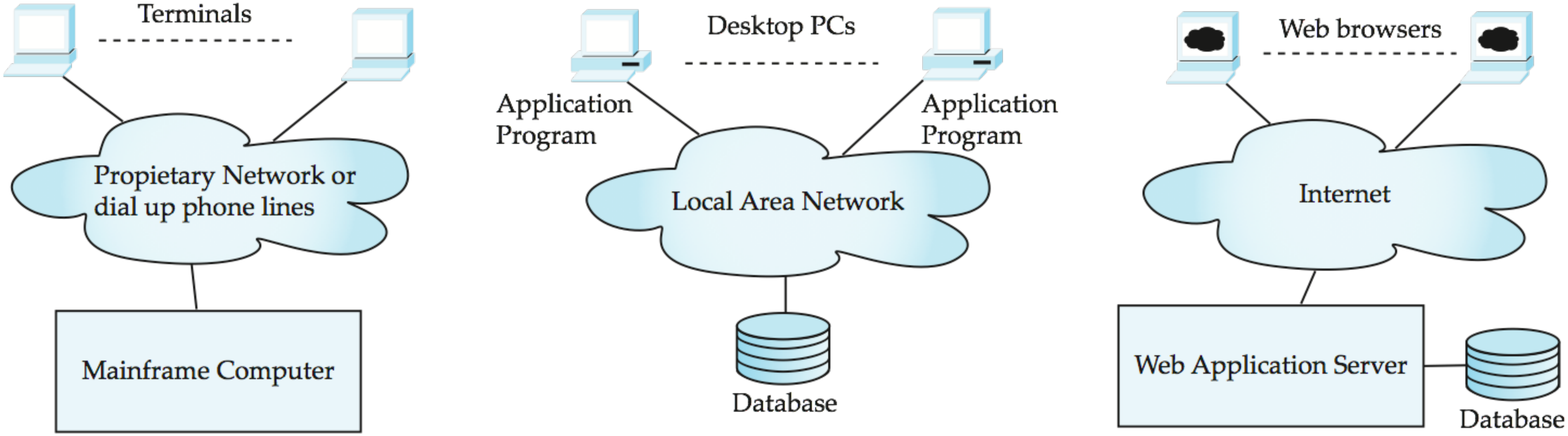
The world before the web

We called them books.



Application Architecture Evolution

- Three distinct era's of application architecture
  - mainframe (1960's and 70's)
  - personal computer era (1980's)
  - We era (1990's onwards)



(a) Mainframe Era

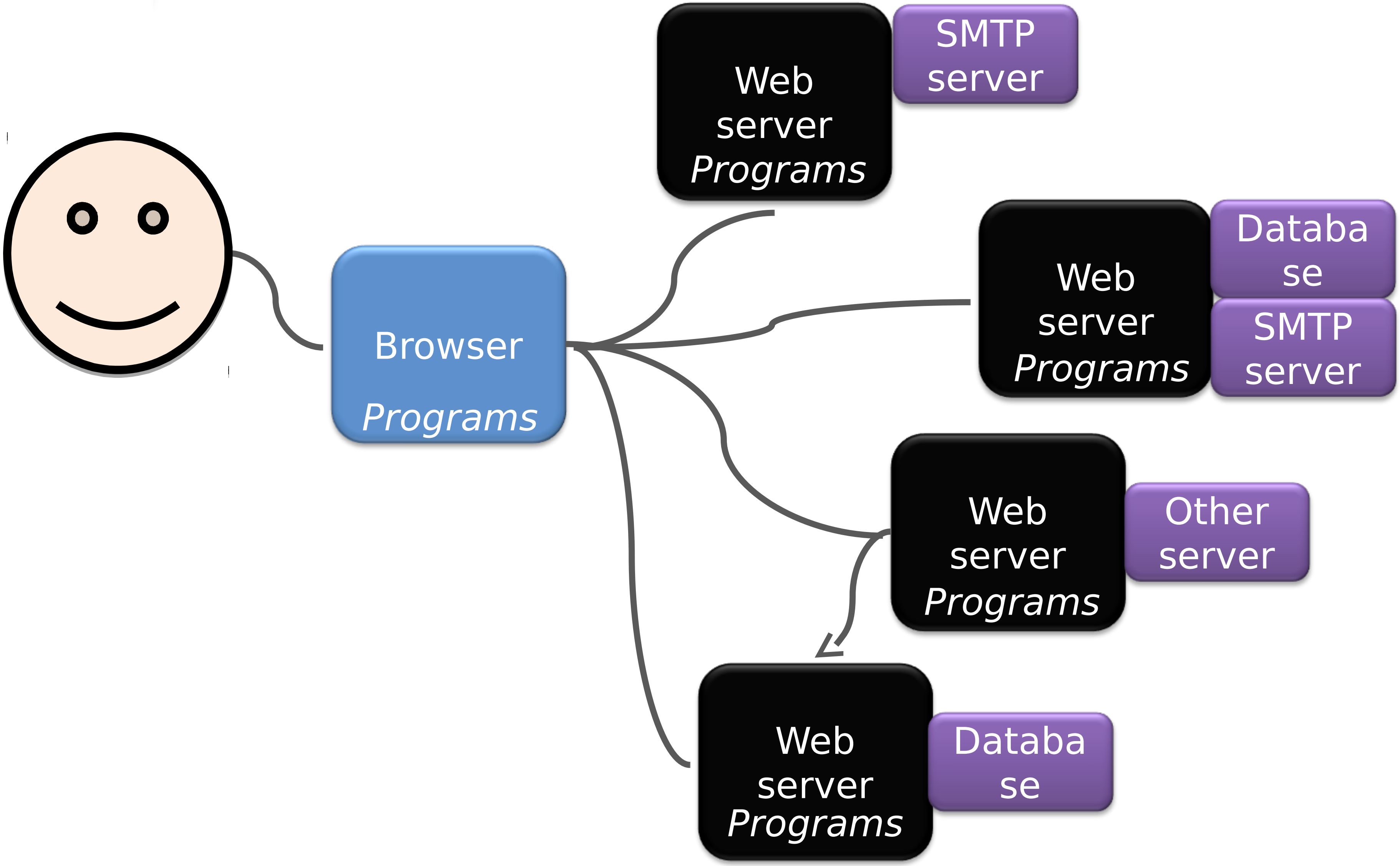
(b) Personal Computer Era

(c) Web era

Your world with the web

- Need some information?
  - Google it. Read on Wikipedia. Ask Yahoo Answers.
- Need to manage your money?
  - Track spending with Mint. Do taxes on TurboTax.
- Want to solve world poverty?
  - Lend money via Kiva. Contribute to Khan Academy
- Want to win friends and influence people?
  - Connect on Facebook. Tweet on Twitter. Post on YouTube

What the web is, put simply.



Application Programs and User Interfaces

- Most database users do *not* use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces
  - Many interfaces are Web-based

Web Interface

- Web browsers have become the de-facto standard user interface to databases
  - Enable large numbers of users to access databases from anywhere
  - Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
    - Javascript, Flash and other scripting languages run in browser, but are downloaded transparently
  - Examples: banks, airline and rental car reservations, university course registration and grading, an so on.
  - HyperText Transfer Protocol (HTTP) used for communication with the Web server

The World Wide Web

- The Web is a distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML)
- HTML documents contain
  - text along with font specifications, and other formatting instructions, images, tables etc.
  - hypertext links to other documents, which can be associated with regions

Uniform Resources Locators

- In the Web, functionality of pointers is provided by Uniform Resource Locators (URLs).
- URL example:

  http://www.acm.org/sigmod

  - The first part indicates how the document is to be accessed
    - “http” indicates that the document is to be accessed using the Hyper Text Transfer Protocol.
  - The second part gives the unique name of a machine on the Internet.
  - The rest of the URL identifies the document within the machine.
- The local identification can be:
    - The path name of a file on the machine, or
    - An identifier (path name) of a program, plus arguments to be passed to the program

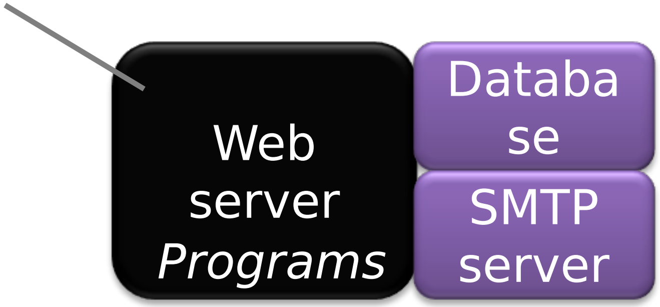
Web Servers

- A Web server can easily serve as a front end to a variety of information services.
- The document name in a URL may identify an executable program, that, when run, generates a HTML document.
  - When an HTTP server receives a request for such a document, it executes the program, and sends back the HTML document that is generated.
  - The Web client can pass extra arguments with the name of the document.
- To install a new service on the Web, one simply needs to create and install an executable that provides that service.
  - The Web browser provides a graphical user interface to the information service.
- Common Gateway Interface (CGI): a standard interface between web and application server

How does it work?

1. Software constantly running, waiting for browsers to connect

6. Web server sends page to browser.



Web server
Programs

Databa
se

SMTP
server

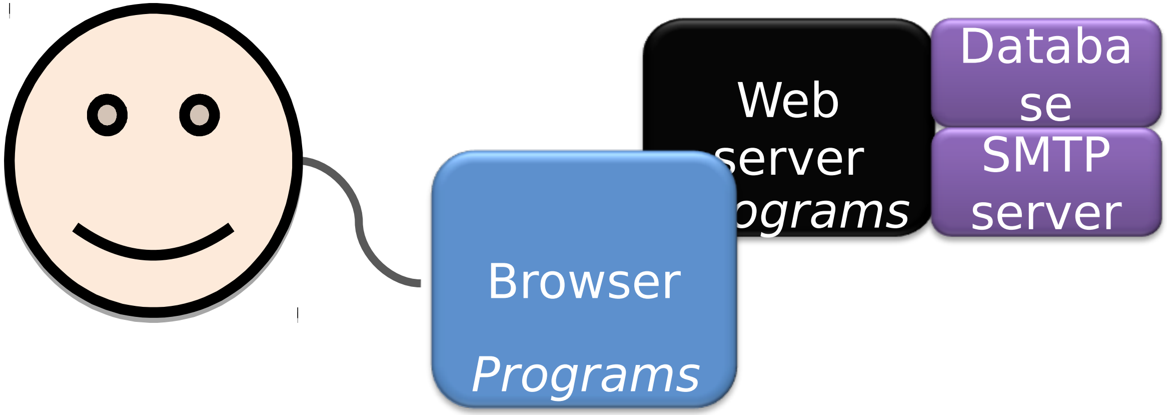
3. Programs read and/or write data stored in the database

4. Programs might also contact other servers (e.g., send email via SMTP)

2. When browser connects, web server sends data from the browser to the programs

5. Program makes a web page and hands it back to the web server.

Web pages can contain programs, too



7. Any programs inside the web page execute.

Primary technologies for the web

- Hypertext markup language (HTML)
- Cascading style sheets (CSS)
- Client side scripting - JavaScript (JS)
- Server side scripting – python, perl,php,asp
- Structured query language (SQL)

Hypertext markup language (HTML)

- This is the language used to “write” web pages

- It describes what is “on” a web page

- HTML is necessary because without it, your web page is empty.

Cascading style sheets (CSS)

- This is the language used to control the appearance of web pages.

- You can control appearance of web pages using just HTML (so CSS is optional).

- But web pages that don’t use CSS are ugly.

JavaScript (JS)

- This is the language used to write the programs than run in the browser.

- JavaScript is awesome because enables web pages to “feel” very interactive.

Server side scripting

- This is the language used to write programs that run on the server.
- Java, Ruby, Python, Perl, Asp, Php and C#.

Structured query language (SQL)

- This is the language used to tell the database what to do (including read and write data).
- There are alternatives, but none of them are as widely supported or standardized.
- SQL (or one of those alternatives) is typically used when you need to manage lots of data

With web technologies, you can build

- Your own **Facebook** – help people make friends and share information
- Your own **Kiva** – help people lend money to others around the world
- Your own **Mint** – help people track their spending and manage their money
- Your own **Google** – help people find information from other web sites

What will you do with your new skills?

- A better way to share information…
  - More specialized than Google. More flexible than Wikipedia and Yahoo Answers.
- Ideas:
  - A website where students can coach one another about which companies give good internships and which ones don’t?

What will you do with your new skills?

- A better way to manage your money…
  - Less intrusive than Mint, less specialized than TurboTax.
- Ideas:
  - A website where you can describe your financial situation, and the website coaches you on financial decisions.
  - A website where you can describe your problems and get real-time guidance from professionals (for free)?

What will you do with your new skills?

- A better way to fight poverty…
  - More personal than Kiva, focused differently than Kiva or Khan.

- Ideas:
  - A website where people can share “tribute” videos that tell stories about people who have touched their lives? Inspire me.

  - A website where student teams can sign up to deliver food to homeless shelters on a schedule.

What will you do with your new skills?

- A better way to win friends and influence people…
  - More private than Facebook, more flexible than Twitter, less work than YouTube

- Ideas:
  - A website where students can challenge each other to sports?

  - A website where students can offer favors (rides?) to one another?

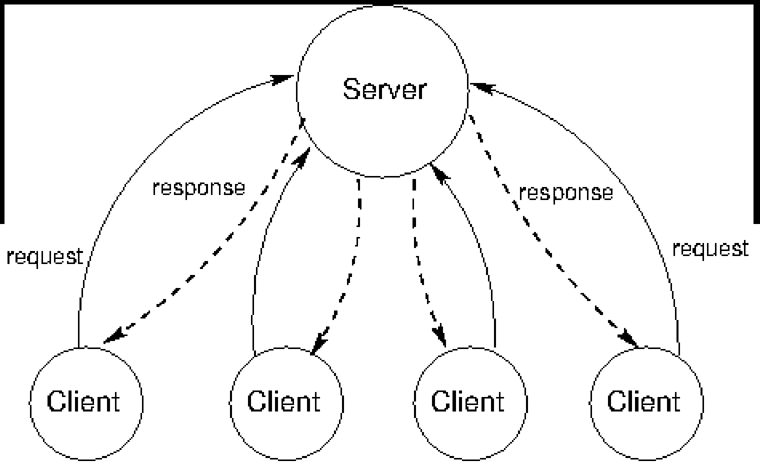
And a billion other possibilities

- New kinds of video games
- New ways to track energy usage
- New methods of learning from mistakes
- New features for planning travel
- New ways to meet people
- New tools for learning useful skills
- **New ways to do anything that matters**

Web Components

- **Clients and Servers**
- **Internet Service Providers**
- **Web Site Hosting Services**
- **Domains Names, URL’s and Ips**
- **Registrars**

Clients & Servers



Clients (Browser)

- Internet Explorer
- Firefox
- Mozilla
- Netscape
- Opera
- Amaya
- AOL
- MSN

Servers

- Apache
- Microsoft
- Netscape
- zeus
- AOLserver
- AV
- JavaWebServer
- Oracle

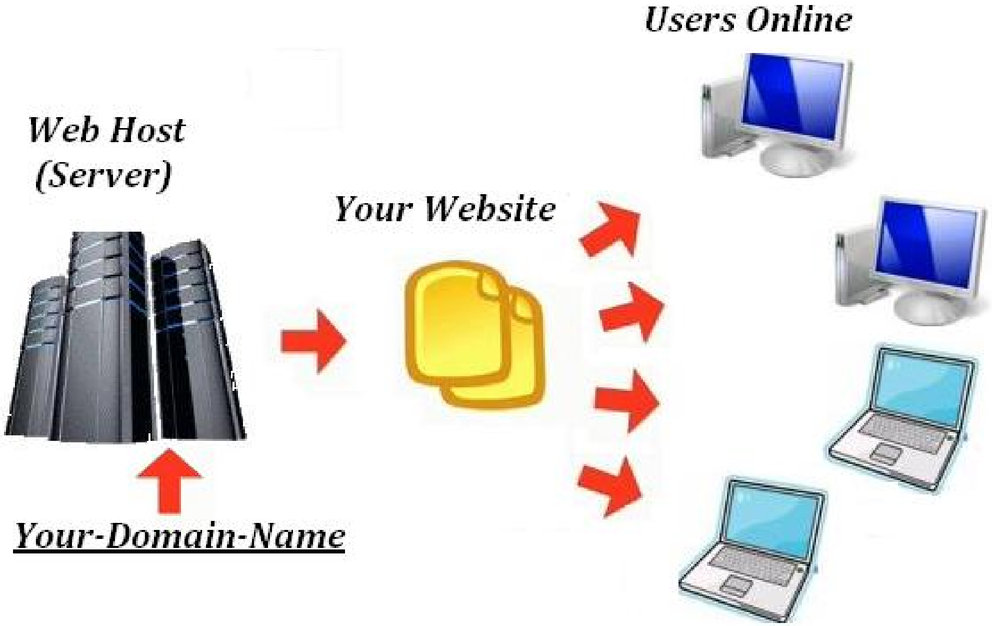
Internet Service Providers

- Phone Company
- AOL
- Earthlink
- Verizone
- NetZero
- Basic internet connection
- Dialup/DSL/Cable/Sat
- Email

Web Hosting Services

Connects Web Sites to the Internet

- Computer (server) farm
- Web server software
- Firewall hardware and software
- IT services
  - (Backup, troubleshooting, hardware repair)
- Disk space
- Bandwidth / connection to internet
- Routers and switchers
- Email server / storage



Domain’s URL’s and IPs

- Domain name: The specific address of a computer on the Internet
  - microsoft.com
- Uniform Resource Locator (URL):
  - http://www.microsoft.com/faqs.html
- Internet protocol (IP) address
  - 192.168.1.1

Domain Registrar

- A company that provides domain name registration services for a fee.
- Maintain database which maps domain names to IP’s
- Propagate new domain name/IP address information across the internet

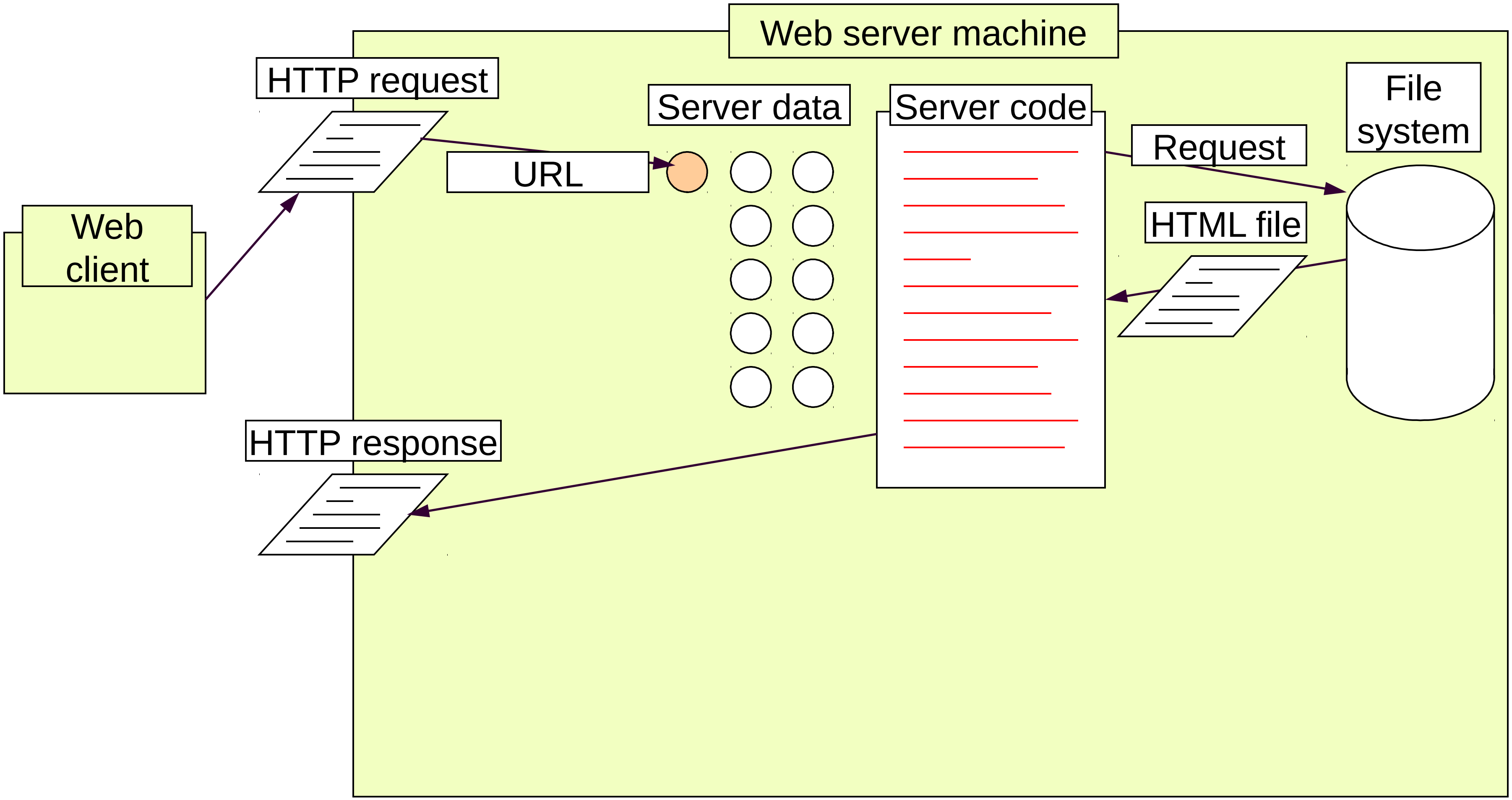
Creating A Web Site

1. Choose a domain name
2. Register with a Registrar
3. Choose a hosting service
4. Tell Registrar the IP address
5. Create web content
6. Store (publish) onto hosting server (FTP)
7. Submit new site to search engines

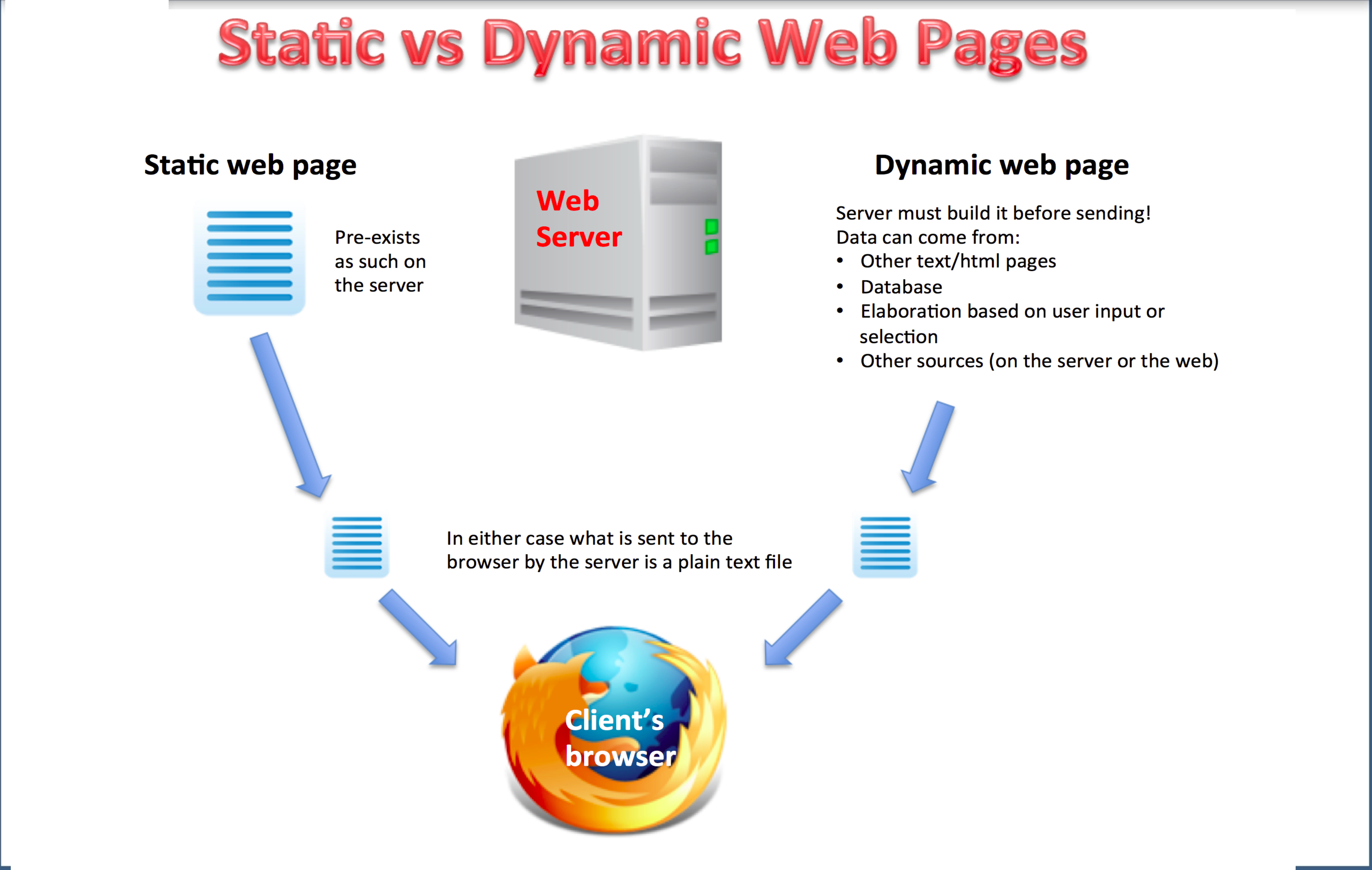
Overview

- Web documents – static and dynamic
- Application Design
- Client-side programming
- Web services
- Security

“Lifecycle” of static web page



Dynamic Web Pages



Static vs Dynamic Web Pages

Static web page

Pre-exists as such on the server

Dynamic web page

Server must build it before sending!
Data can come from:

- Other text/html pages
- Database
- Elaboration based on user input or selection
- Other sources (on the server or the web)

In either case what is sent to the browser by the server is a plain text file

Dynamic Web Pages

2. Dynamic generation of documents

- Limitations of static HTML documents
  - Cannot customize fixed Web documents for individual users.
  - Problematic to update Web documents, especially if multiple Web documents replicate data.
- Solution: Generate Web documents dynamically from data stored in a database.
  - Can tailor the display based on user information stored in the database.
    - E.g., tailored ads, tailored weather and local news, ...
  - Displayed information is up-to-date, unlike the static Web pages
  - E.g., stock market information, ..

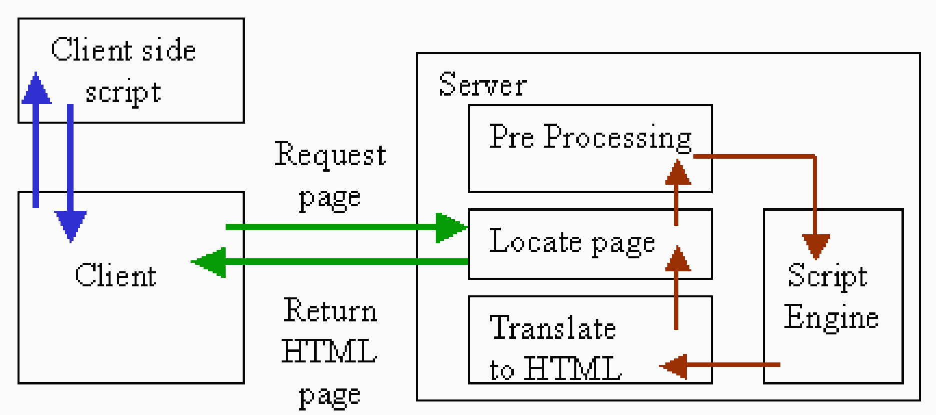
Client Side Programming

Web pages can contain JavaScript programs executed inside the browser

- Supported by all major browsers
  - Microsoft’s version called Jscript (the language is the same)
- User may disable JavaScript due to security fears
  - This is default for some newer versions of Internet Explorer

Client-side scripts/programs allow documents to be active

- Can promptly **validate user input**
- **animation** by executing programs at the local site
- Can update the web page without postback to server and hence **reduced round trips to server**.
- Allows page to react to user actions other than pushing a “submit” button – **more interactivity**



COMPARISON

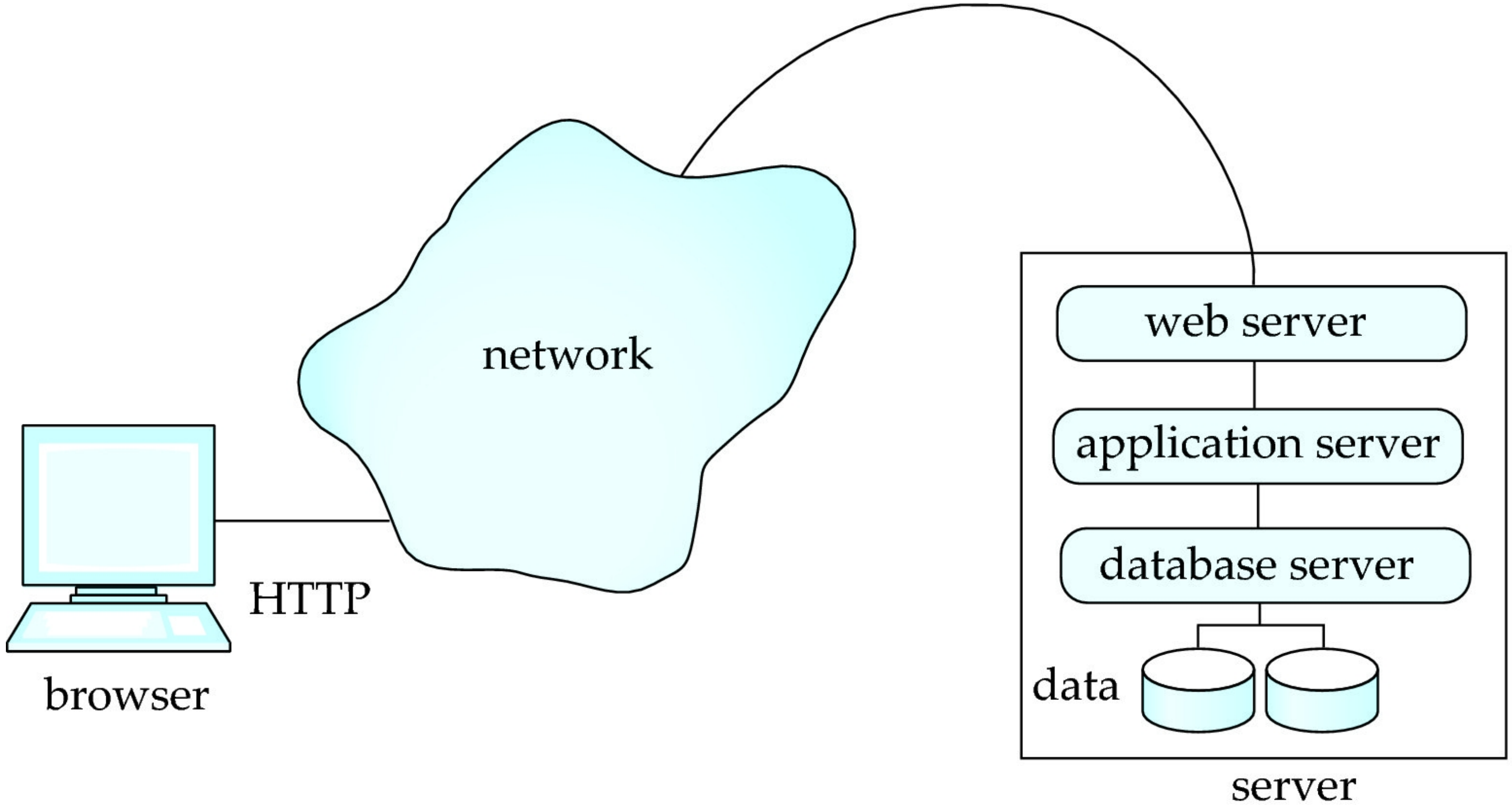
Client side scripting

- Used when the users browser already has all the code
- The Web Browser executes the client side scripting
- Cannot be used to connect to the databases on the web server
- Can't access the file system that resides at the web server
- Response from a client-side script is faster as compared to a server-side script

Server side scripting

- Used to create dynamic pages
- The Web Server executes the server side scripting
- Used to connect to the databases that reside on the web server
- Can access the file system residing at the web server
- Response from a server-side script is slower as compared to a client-side script

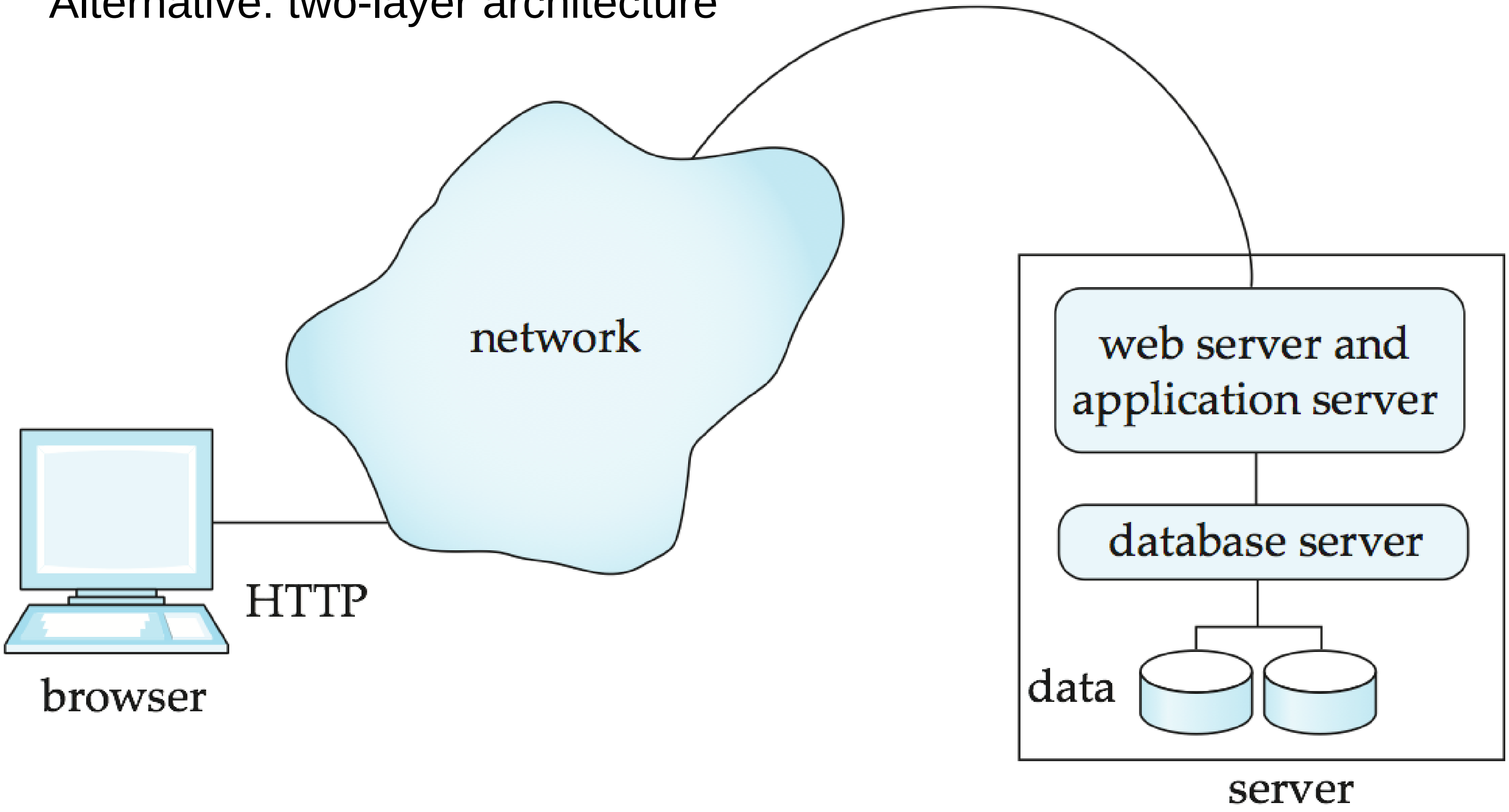
Three-Layer Web Architecture



Two-Layer Web Architecture

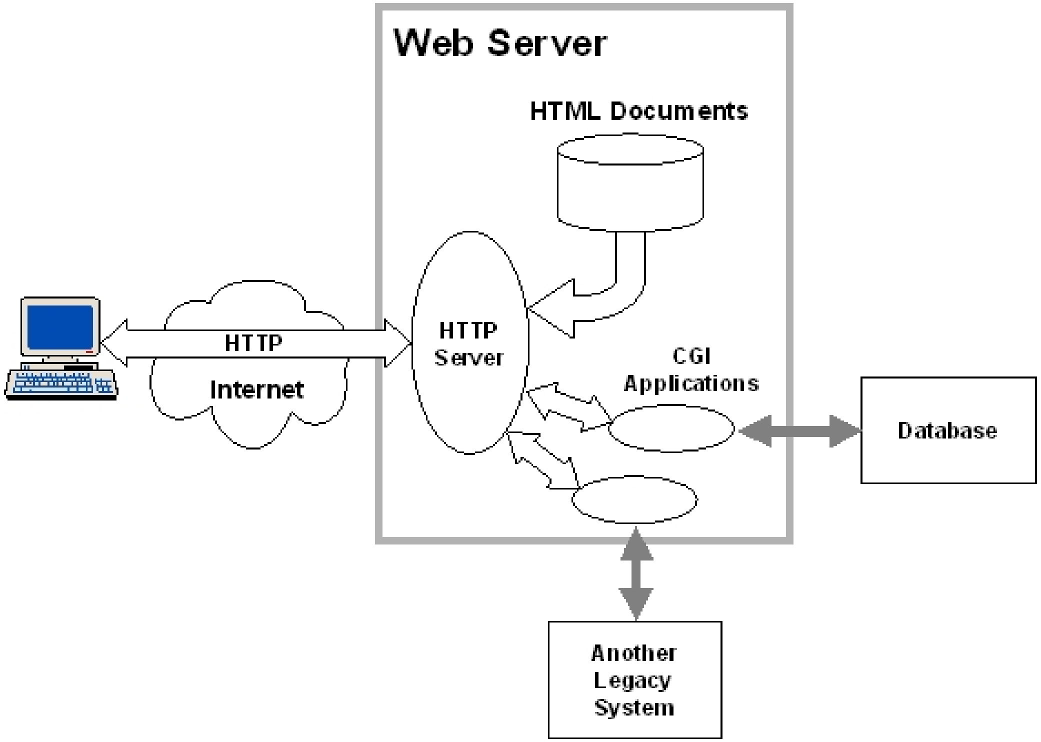
Multiple levels of indirection have overheads

- Alternative: two-layer architecture



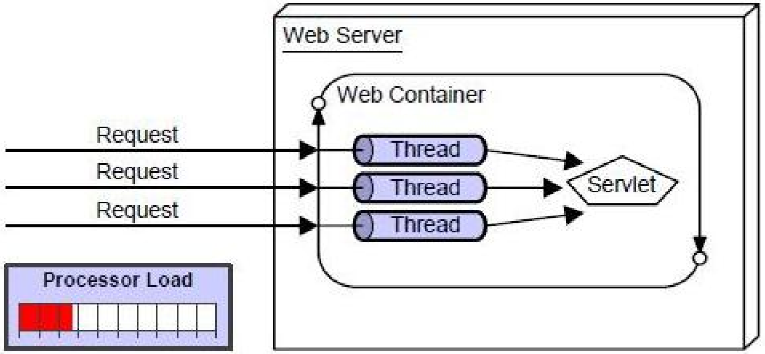
Server side programming

- CGI – separate programs launched by web server,
- Strong isolation, bad performance
- Programs embedded inside web page (php, ASP, JSP, python)
    - F rocess



Servlets

- Application program (also called a servlet) is loaded into the server
  - Each request spawns a new thread in the server
    - thread is closed once the request is serviced



Server-Side Scripting

- Server-side scripting simplifies the task of connecting a database to the Web
  - Define an HTML document with embedded executable code/SQL queries.
  - Input values from HTML forms can be used directly in the embedded code/SQL queries.
  - When the document is requested, the Web server executes the embedded code/SQL queries to generate the actual HTML document.
- Numerous server-side scripting languages
  - JSP, PHP
  - General purpose scripting languages: VBScript, Perl, Python

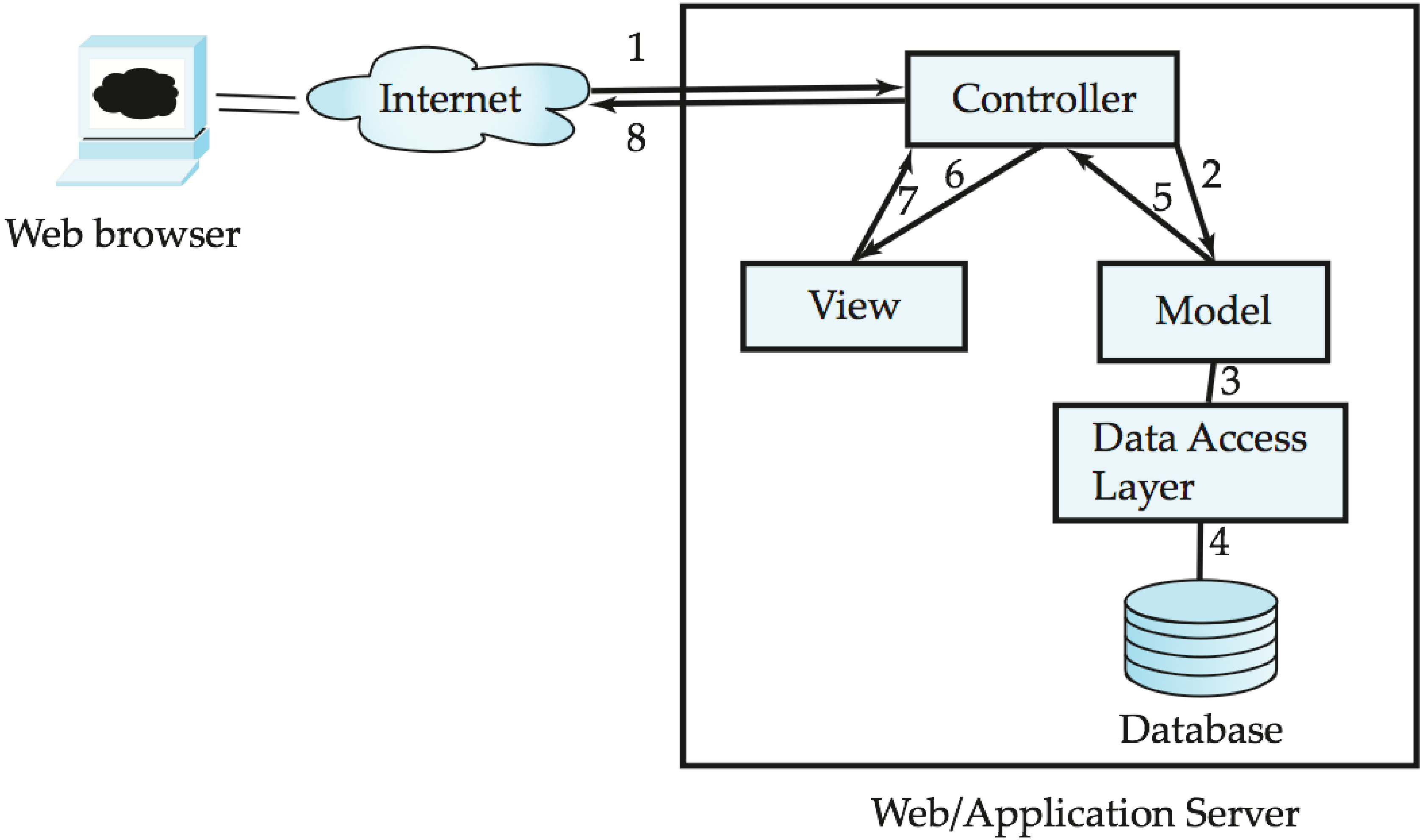
Client Side Scripting and Security

- Security mechanisms needed to ensure that malicious scripts do not cause damage to the client machine
  - Easy for limited capability scripting languages, harder for general purpose programming languages like Java
- E.g., Java’s security system ensures that the Java applet code does not make any system calls directly
  - Disallows dangerous actions such as file writes
  - Notifies the user about potentially dangerous actions, and allows the option to abort the

Application Architectures

- Application layers
  - Presentation or user interface
    - **model-view-controller (MVC)** architecture
      - **model**: business logic
      - **view**: presentation of data, depends on display device
      - **controller**: receives events, executes actions, and returns a view to the user
  - **business-logic** layer
    - provides high level view of data and actions on data
      - often using an object data model
    - hides details of data storage schema
  - **data access** layer
    - interfaces between business logic layer and the underlying database
    - provides mapping from object model of business layer to relational model of database

Application Architecture



Business Logic Layer

- Provides abstractions of entities
  - e.g. students, instructors, courses, etc
- Enforces **business rules** for carrying out actions
  - E.g. student can enroll in a class only if she has completed prerequsites, and has paid her tuition fees
- Supports **workflows** which define how a task involving multiple participants is to be carried out
  - E.g. how to process application by a student applying to a university
  - Sequence of steps to carry out task
  - Error handling
    - e.g. what to do if recommendation letters not received on time
  - Workflows discussed in Section 26.2