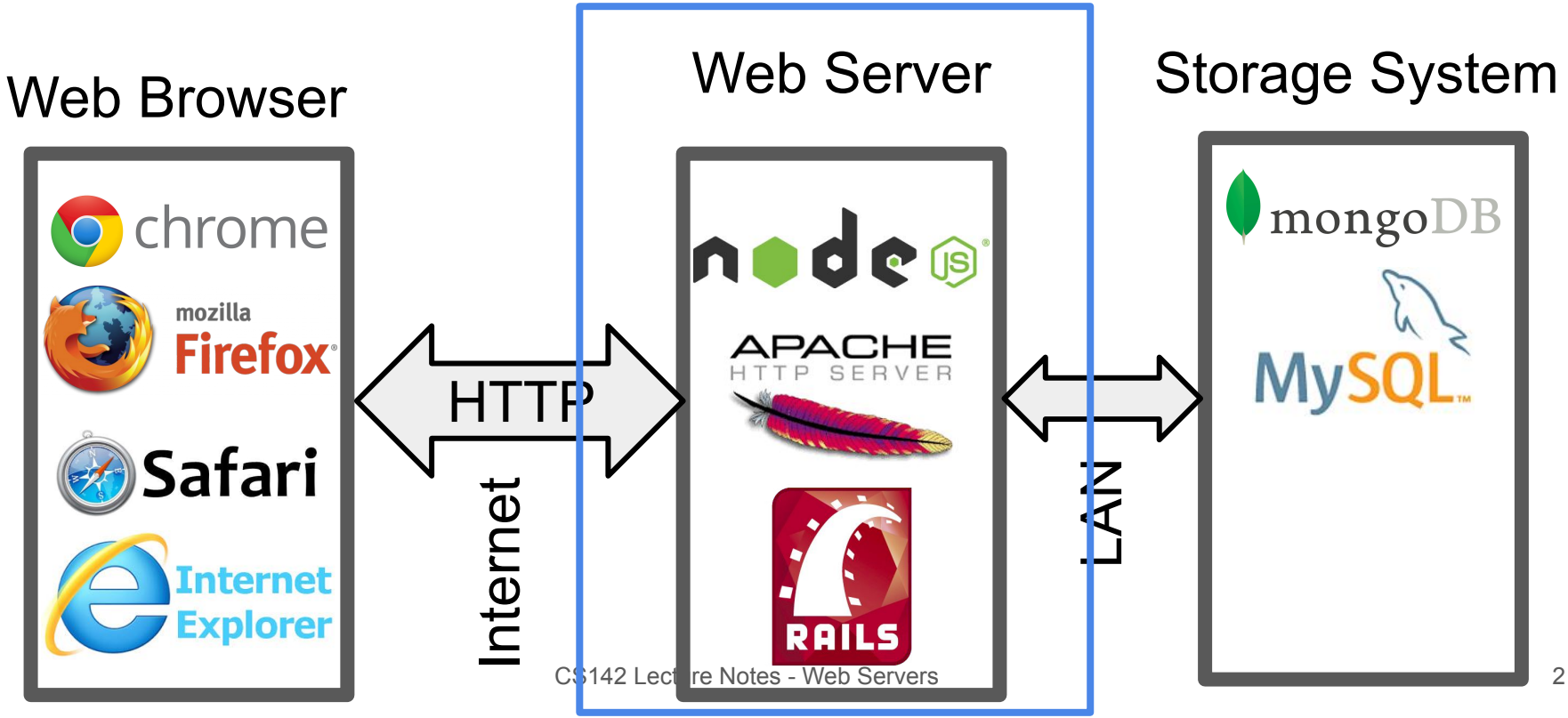
Web Servers

Mendel Rosenblum

Web Application Architecture



Web Servers

- Browsers speak HTTP and Web Servers speak HTTP
  - Browsers: send HTTP request and get HTTP responses
  - Web Server: get HTTP requests and send HTTP responses
- HTTP is layered on TCP/IP so a web server:

```
loop forever doing:
    accept TCP connection from browser
    read HTTP request from TCP connection
    process HTTP request
    write HTTP response to TCP connection
    shutdown TCP connection (except if Connection: keep-alive)
```

Processing HTTP requests - File reads

- Process HTTP GET index.html

```
int fd = open("index.html");
int len = read(fd, fileContents, sizeOfFile(fd));
write(tcpConnection, httpResponseHeader, headerSize);
write(tcpConnection, fileContents, len);
```

- Note open and read may have to talk to a slow disk device
  - Can process requests concurrently by starting a new thread or a new process per request

Processing HTTP requests - cgi-bin

- Process HTTP GET of index.php

  ```
  runProgramInNewProcess(tcpConnection);
  ```

- Template processing program fetches models from database system

2nd Generation Web App Frameworks

Web server runs a program per request - the **controller**:

1. Parse URL and/or HTTP request body to get parameters to view
2. Use parameters to fetch **model** data from DBMS (typically a SQL relational DBMS)
3. Run HTML **view** template with model data to generate the HTML
4. Send a HTTP response with the HTML back to the browser

Rails runs a controller program per URL. Example: URL `/rails_intro/hello`

Runs controller `hello.rb` (Ruby program fetches models - ORM)

Applies to view template `hello.html.erb` (HTML embedded with Ruby)

JavaScript?: An asset (like an image or css) you can include

Web servers for JavaScript frameworks

- Most of the web app is simple static files - any web server speaking HTTP
  - View templates (HTML, CSS)
  - JavaScript files
- Remaining browser⇔ server communication around model data
  - CRUD (Create Read Update Delete) of model data
  - Session info (e.g. login, etc.) (Later…)
- Low requirements on web request processing
  - HTTP GET static files
  - Model data operation - mostly doing DBMS operations