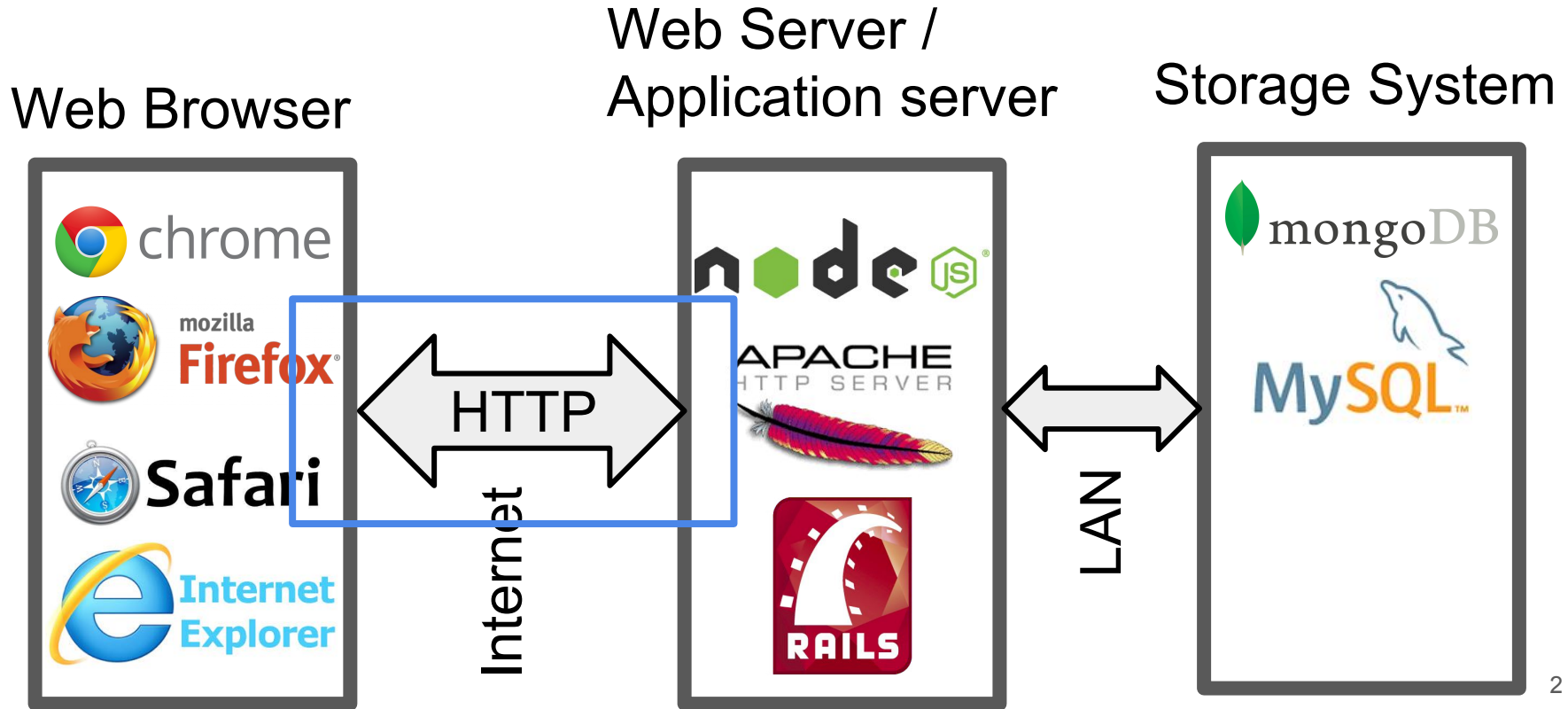


Hypertext Transport Protocol (HTTP)

Mendel Rosenblum

Web Application Architecture



http://www.example.com:80/index.html

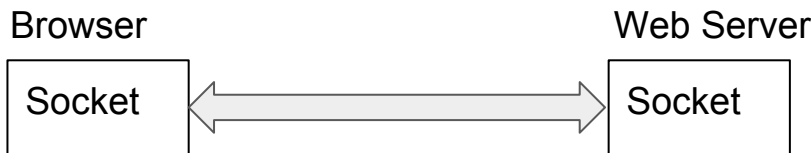
- To display page browser fetches the file `index.html` from a web server
Same as `www.example.com` (Defaults: port 80, file `index.html`, http protocol)
- HTTP (**H**yper**T**ext **T**ransport **P**rotocol)
- HTTP - Simple request-response **protocol** layered on TCP/IP
 1. Establish a **TCP/IP connection** to `www.example.com:80`
 2. Send a http GET **request** along connection
 3. Read from the connection the **response** from the web server

TCP/IP connection to `www.example.com:80`

- TCP/IP - **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol

The Internet: Reliable, in-order, byte-stream protocol

Uses networking socket endpoint abstraction



- Browser needs to convert `www.example.com` to an IP address

Uses DNS (**D**omain **N**ame **S**ervice) to lookup `www.example.com`

Lookup of `www.example.com` returns IP address: `93.184.216.34`

Browser connects to **`93.184.216.34 port 80`**

Send HTTP Request - Write lines to socket

Method URL Protocol Version

↓ ↓ ↓

Header { GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html, */*
Accept-Language: en-us
Accept-Charset: ISO-8859-1,utf-8
Connection: keep-alive

blank line

Body (optional) {

HTTP Methods (Verbs)

- GET - Fetch a URL
- HEAD - Fetch information about a URL
- PUT - Store to an URL
- POST - Send form data to a URL and get a response back
- DELETE - Delete a URL

GET and POST (forms) are commonly used.

REST APIs used GET, PUT, POST, and DELETE

HTTP Response - Read lines from socket

Version Status Status Message

↓ ↓ ↙

Header { HTTP/1.1 200 OK
Date: Thu, 24 Jul 2008 17:36:27 GMT
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=UTF-8
Content-Length: 1846
blank line

Body { <?xml ... >
<!DOCTYPE html ... >
<html ... >
...
</html>

Common HTTP Response Status Codes

200 OK	Success
307 Temporary Redirect	Redirection - Browser retries using Location header
404 Not Found	Famous one
503 Service Unavailable	Something crashed on the server
500 Internal Server Error	Something is messed up on the server
501 Not Implemented	Coming
400 Bad Request	Use if web app sends bogus request
401 Unauthorized	Use if user isn't logged in
403 Forbidden	Use if even logging in wouldn't help
550 Permission denied	Not allow to perform request

Error 404

Whoops. Looks like this page doesn't exist.

404 NOT FOUND

Sorry, this page isn't available
The link you followed may be broken, or the page may have been removed.



Go back to the previous page. Go to the Facebook homepage. Visit the News Center.

404

Not Found



404. That's an error.
The requested URL /asdfasdf was not found on this server. That's all we know.



Not Found

The requested URL /somefolder was not found on this server.
Additionally, a 404 Not Found error was encountered while trying to use an ErrorDocument to handle the request.

404 error

Page not found

We are sorry but the page you are looking for does not exist.
You could return to the homepage or search using the search box below.



The page you're looking for went flying!
Go back to www.frankenbitch.be

404 - Not found

Not Found

The requested URL /index.html was not found on this server.

Apache/1.3.37 Server at www.this-is-a-test.net Port 80

404

Page not found

404

Page not found



CRAP... You have requested the page that is no longer there.

IT'S A 404

SORRY BUDDY YOU ARE FUCKIN' LOST



404

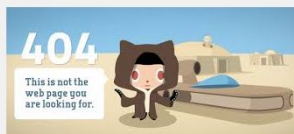
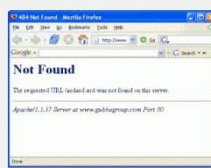
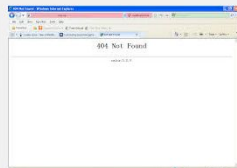
NOT FOUND



ERROR #404



IF YOU LACK OF
NAVIGATION TEST USING



404 NOT FOUND



I think we both know you're in the wrong place...
Why don't you try another door, please?



HTTP Error 404

404 Not Found

The Web server cannot find the file or script you asked for. Please check the URL to ensure that the path is correct.

Please contact the server's administrator if this problem persists.



404
OMG, page was not found



Browser caching control

HTTP Response Header: `Cache-Control: max-age=<Seconds>`

Browser can reuse reply younger than the max-age

`Cache-Control: max-age=120` - Age out in two minutes.

Frequently used on fetches of static content like images, templates, CSS, JavaScript.

Good: Reduce app startup latency and server load

Bad: Changes might not be picked up right away

Consider Web App changes?

Browser spends its life fetching things using HTTP

- Some fetched immediately

```
<link href="angular-material.css" rel="stylesheet" />  
<script src="angular.js" type="text/javascript" ></script>  
window.location = "http://www.example.com";
```

- Some asynchronous and in parallel

```
  
  

```

- Some can be in background

```
<a href="http://www.example.com"></a>
```

What would this JavaScript do?

```
elm.innerHTML =  
    "<script src='http://www.example.com/myJS.js'  
        type='text/javascript' ></script>"
```

Uses HTTP to fetch myJS.js and runs it! Scary but useful.