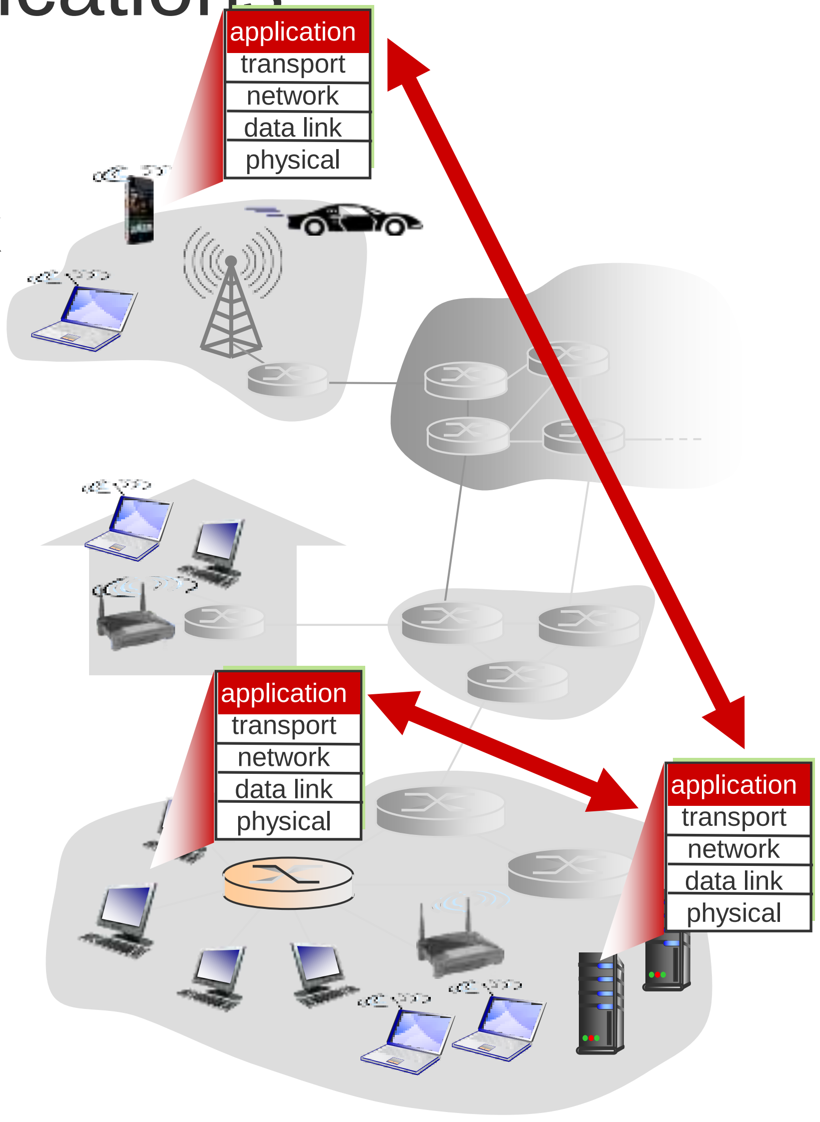
Network Applications (2)

Andy Carpenter

(Andy.Carpenter@manchester.ac.uk)

Network Applications

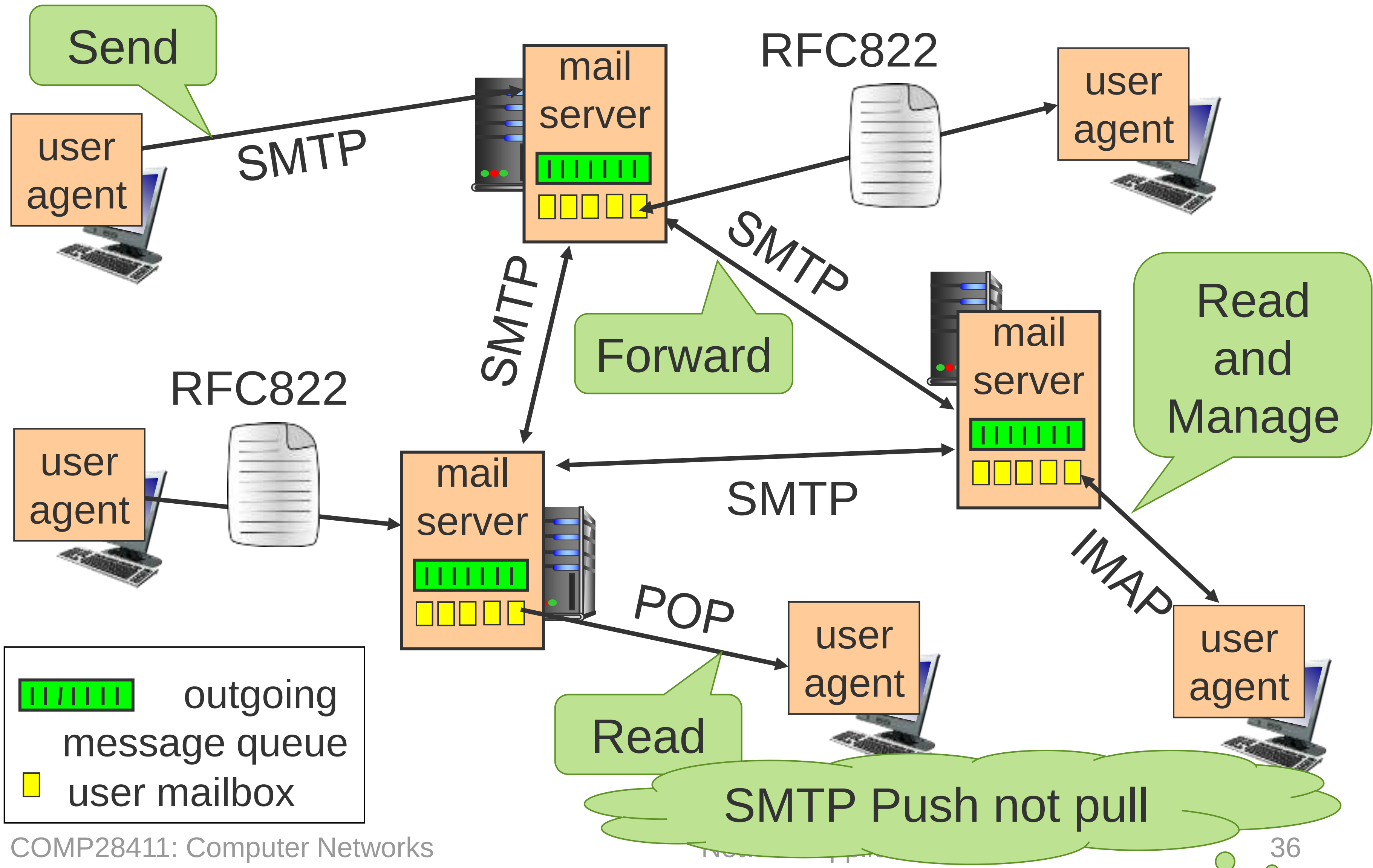
- Programs that:
  - communicate over network
  - run on end systems
- Issues:
  - architecture
  - QoS
  - protocols, addressing
  - understanding data
  - control vs. data
  - extensibility, scalability
  - buffering, state



Application Protocols and Formats

- Protocols exchanges commands between end-points
- Protocols ensure QoS requirements are implemented
- Protocols define:
  - types of message exchanged; e.g. request
  - message syntax; fields present and their delineation
  - message semantics; meaning of fields
  - message exchange rules
- Formats define:
  - structure and meaning of protocol data
- Need single world-wide interpretation; global standards

Protocols: Electronic Mail



Protocols: Email (SMTP)

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
                 etchup?
                 les?

S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Establish

Commands

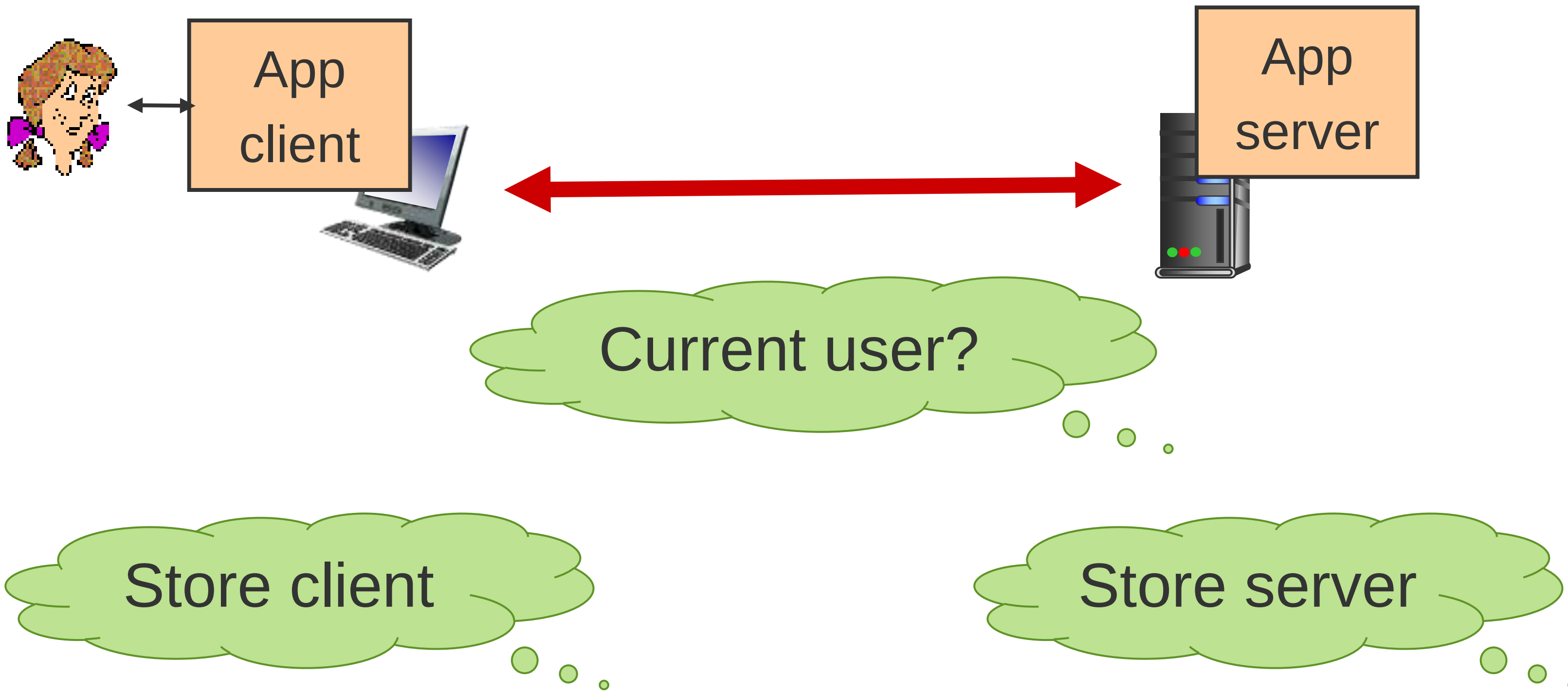
Envelope for message

Data

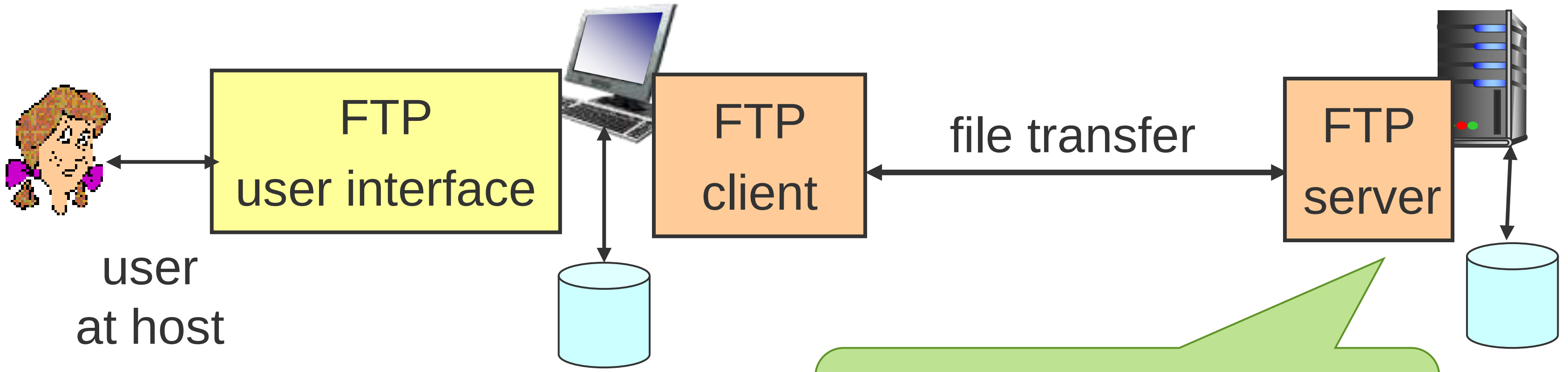
Close

Status code and phrase

Application State



State: FTP

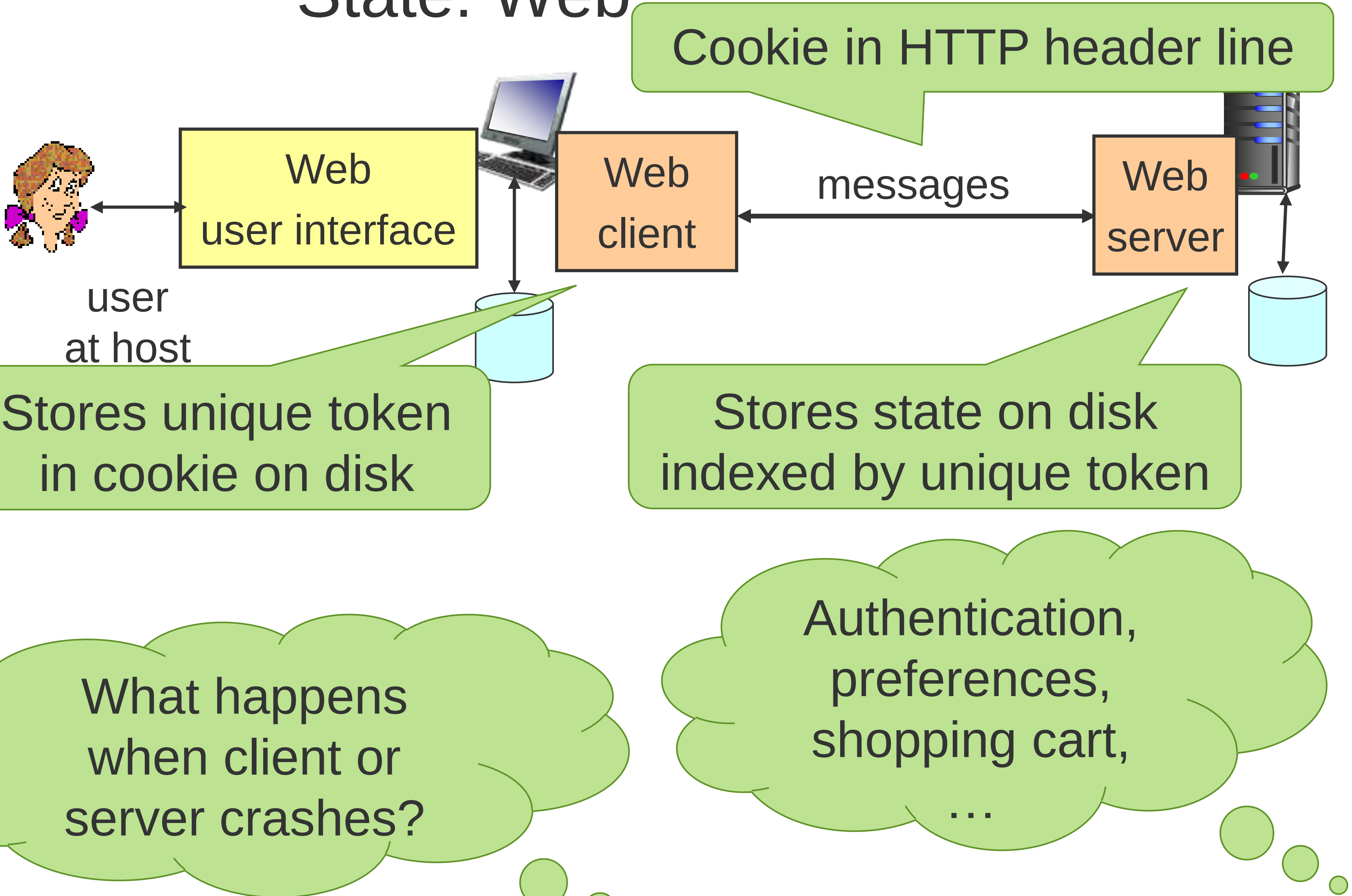


Stores state in server process

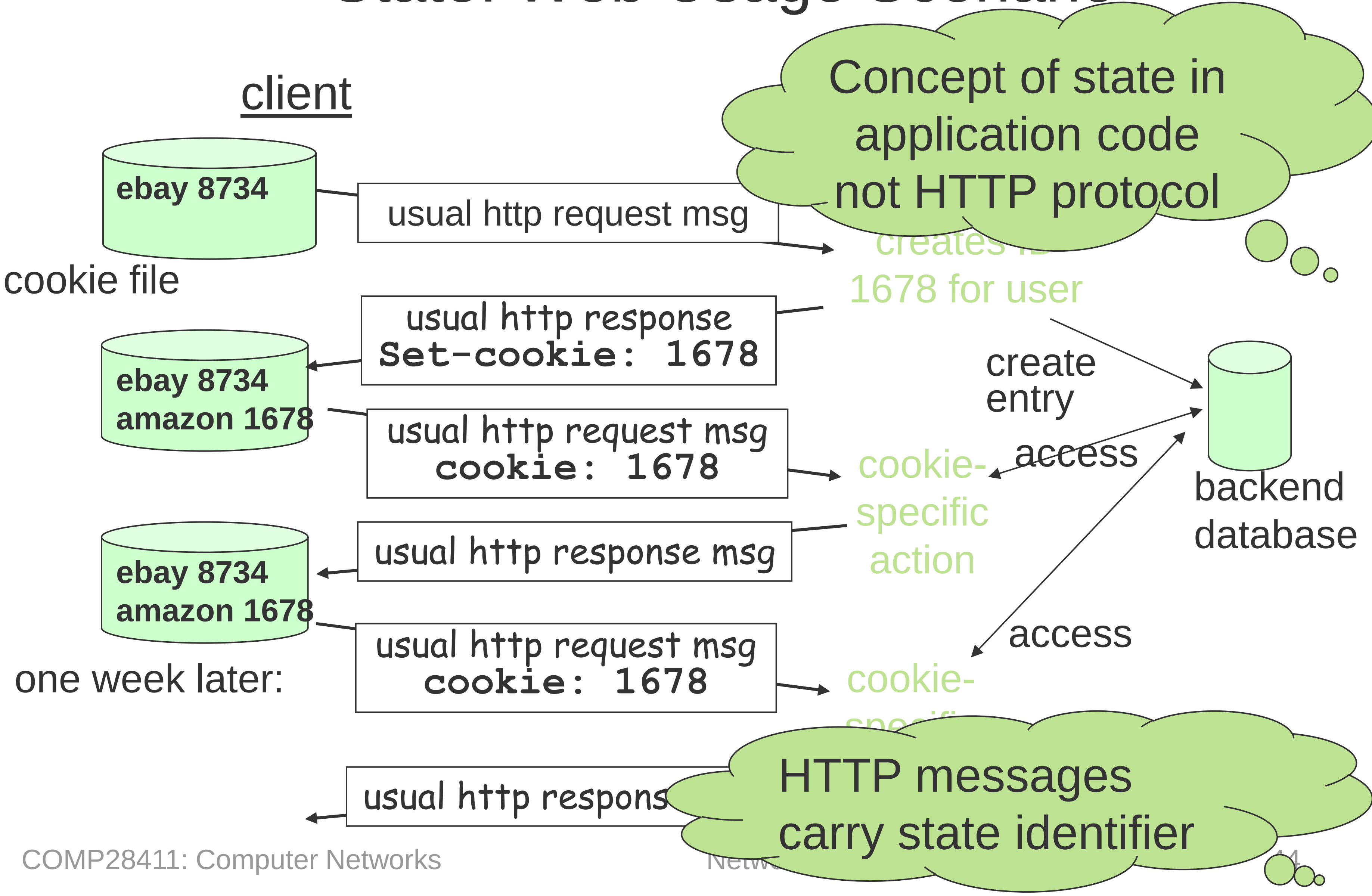
What happens when client or server crashes?

Authentication, current directory, …

State: Web


Cookie in HTTP header line
Web user interface
Web client
messages
Web server
user at host
Stores unique token in cookie on disk
Stores state on disk indexed by unique token
What happens when client or server crashes?
Authentication, preferences, shopping cart, …

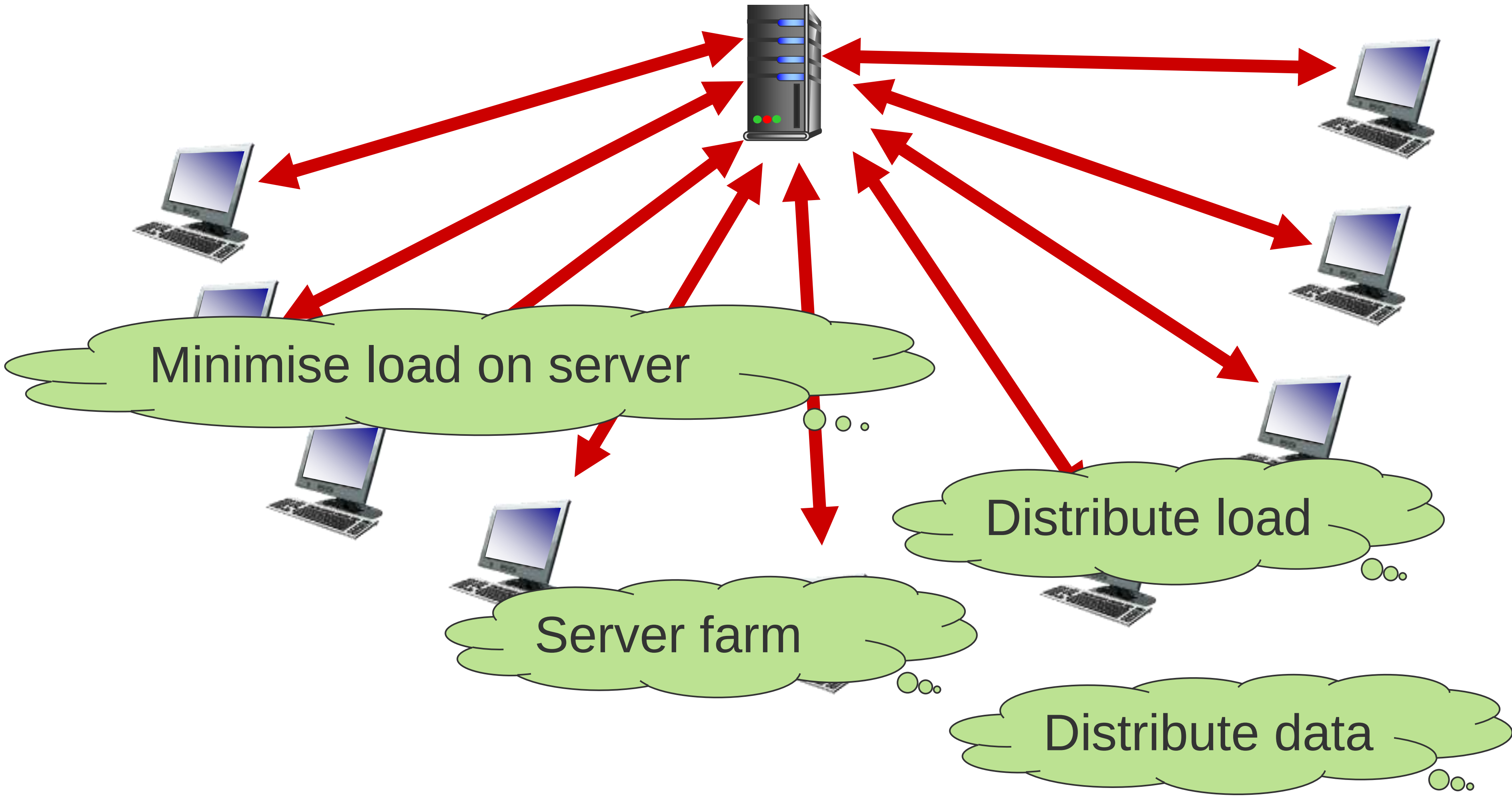
State: Web Usage Scenario



Concept of state in application code not HTTP protocol

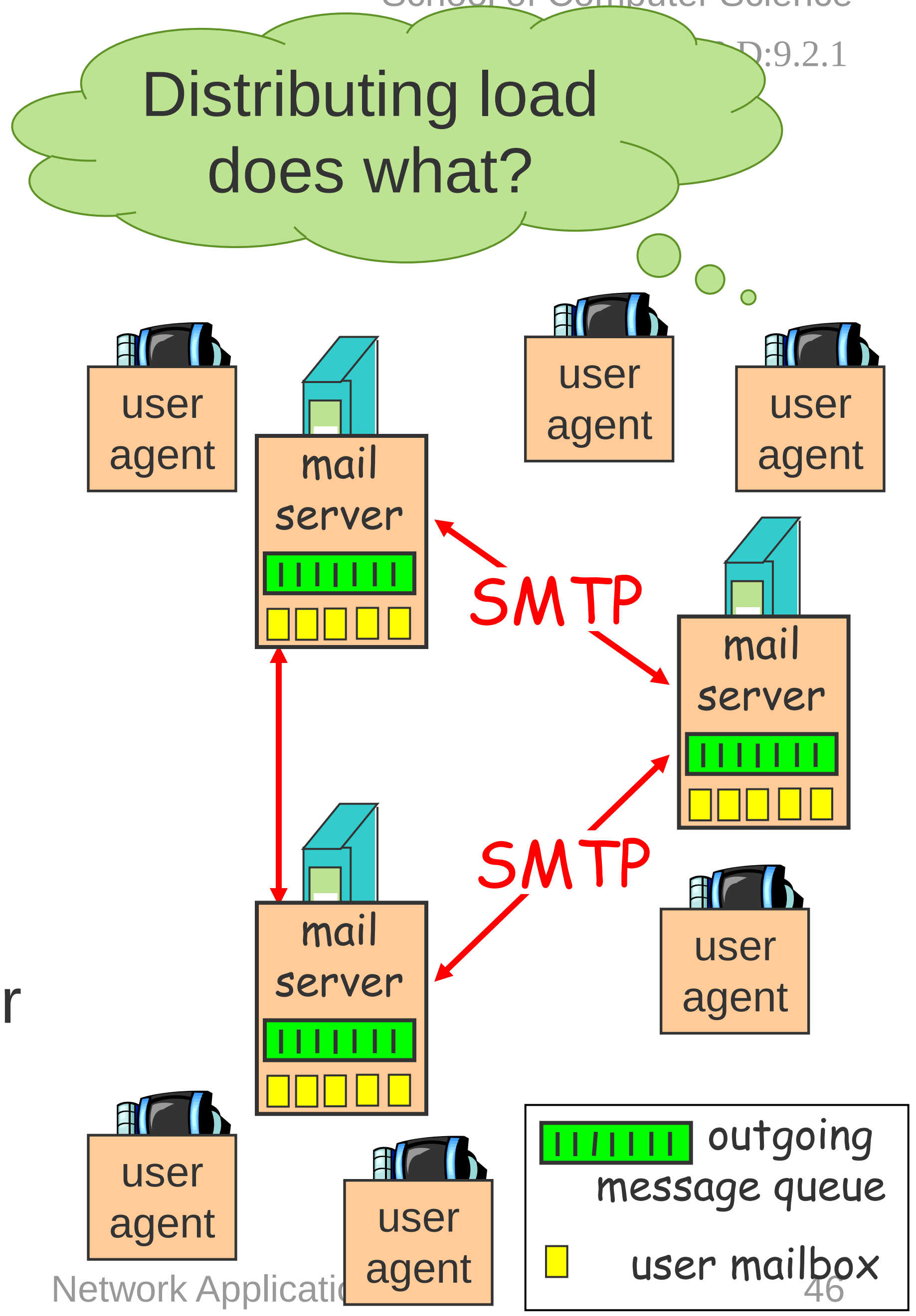
HTTP messages carry state identifier

Application Scalability

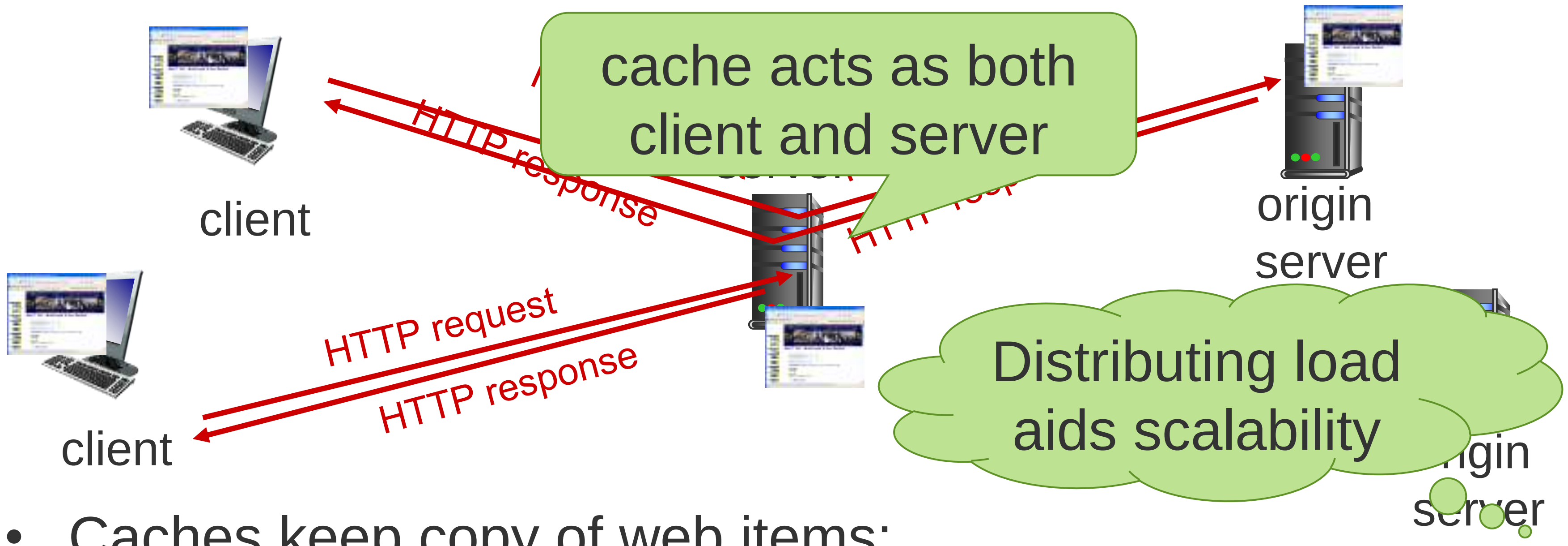


Email: Mail Servers

- Mailbox per user contains incoming messages
- Message queue of (unsent) outgoing messages
- Email messages between servers sent using SMTP
  - client: sending server
  - "server": receiving server

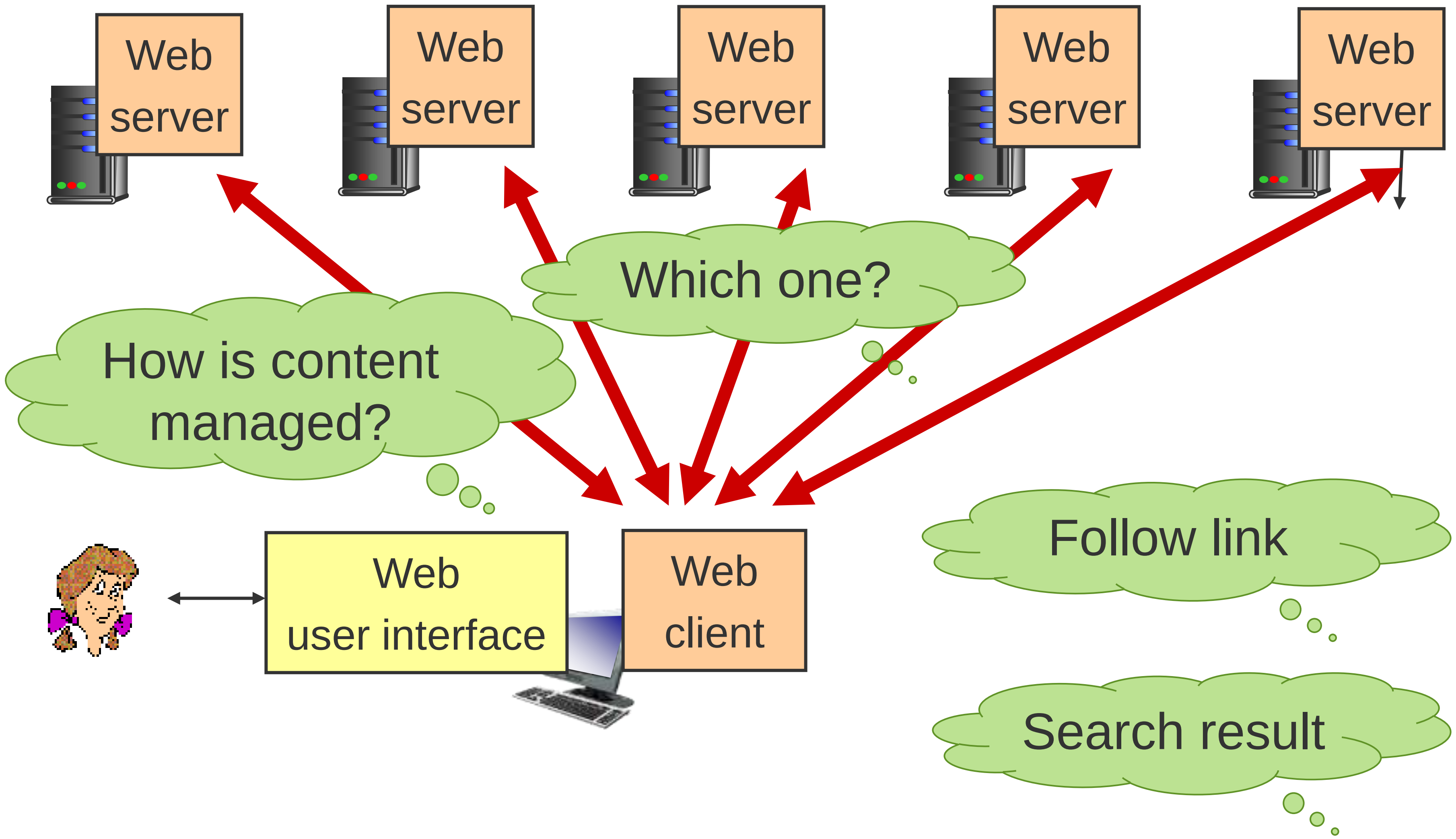


Scalability: Web Caching



- Caches keep copy of web items:
  - avoids re-fetching, faster accessing
  - reduces network traffic and load on server
- Issue is controlling cache
  - do not responding with out-of-date data
  - response header fields also control caching

Scalability: Web Data Distribution



Scalability: Domain Name System (DNS)

- Uses hierarchical name space for internet objects
- Provides way to decentralise:
  - naming, name and value mapping, resolving
- Not just names to address mapping; others:
  - host and mail server aliases (service names)
  - address to name
  - load balancing (multiple address for one name)
- Issues:
  - coordinated decentralisation, scalability
  - robustness, start point for searches

Scalable

Scalability: DNS Naming

- Hierarchical names; levels separated by a dot
- At top there is a single ‘root’ domain; ‘.’
- A section of hierarchy is known as a domain or zone
- Names must be unique within a zone
- Standard assumes top level naming authority
- May delegate naming authority for a domain/zone
- Naming authority may be further delegated
- Domains are divided until contents are manageable

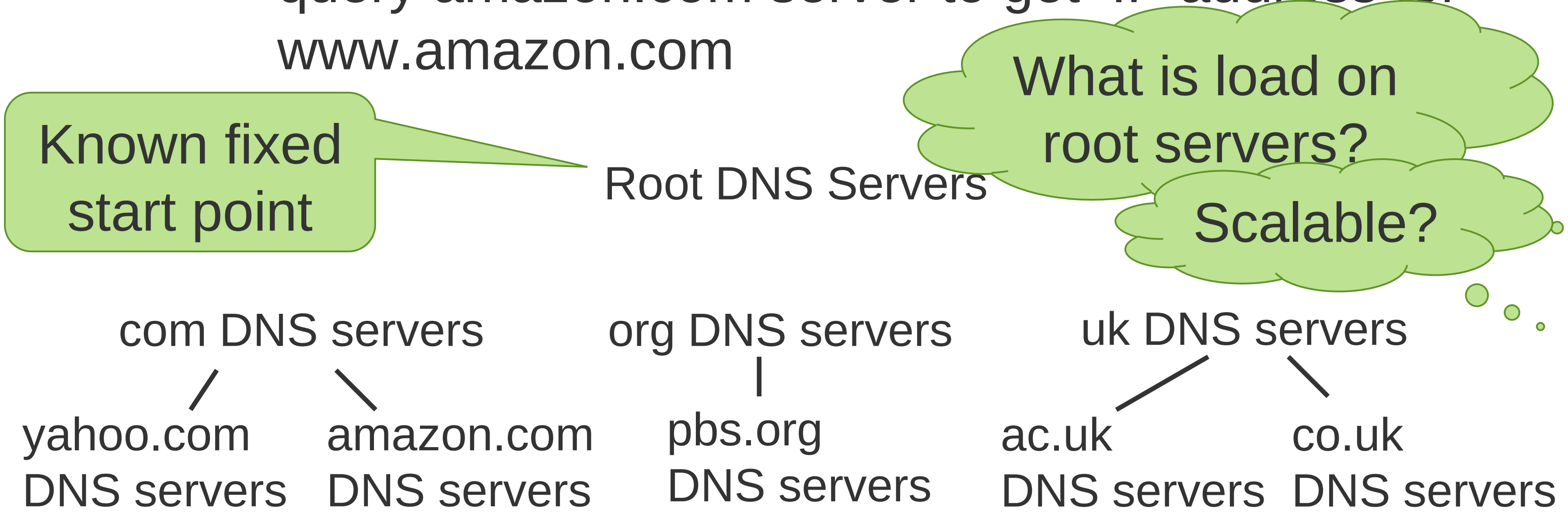
Distributed management

Scalable

How is uniqueness ensured?

Scalability: DNS ‘Database’

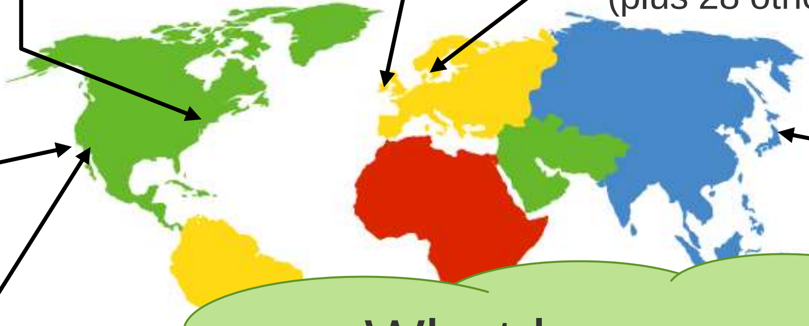
- Every zone has, at least, one name server
- Client wants IP for www.amazon.com; 1st approx:
  – query a root server to find com DNS server
  – query com server to get amazon.com DNS server
  – query amazon.com server to get IP address for www.amazon.com



DNS: Fixed Start Points for Queries

- 13 root name servers worldwide
  - named [a-m].root-servers.net
  - many have multiple locations (use anycasting)
- Addresses built into DNS implementation code

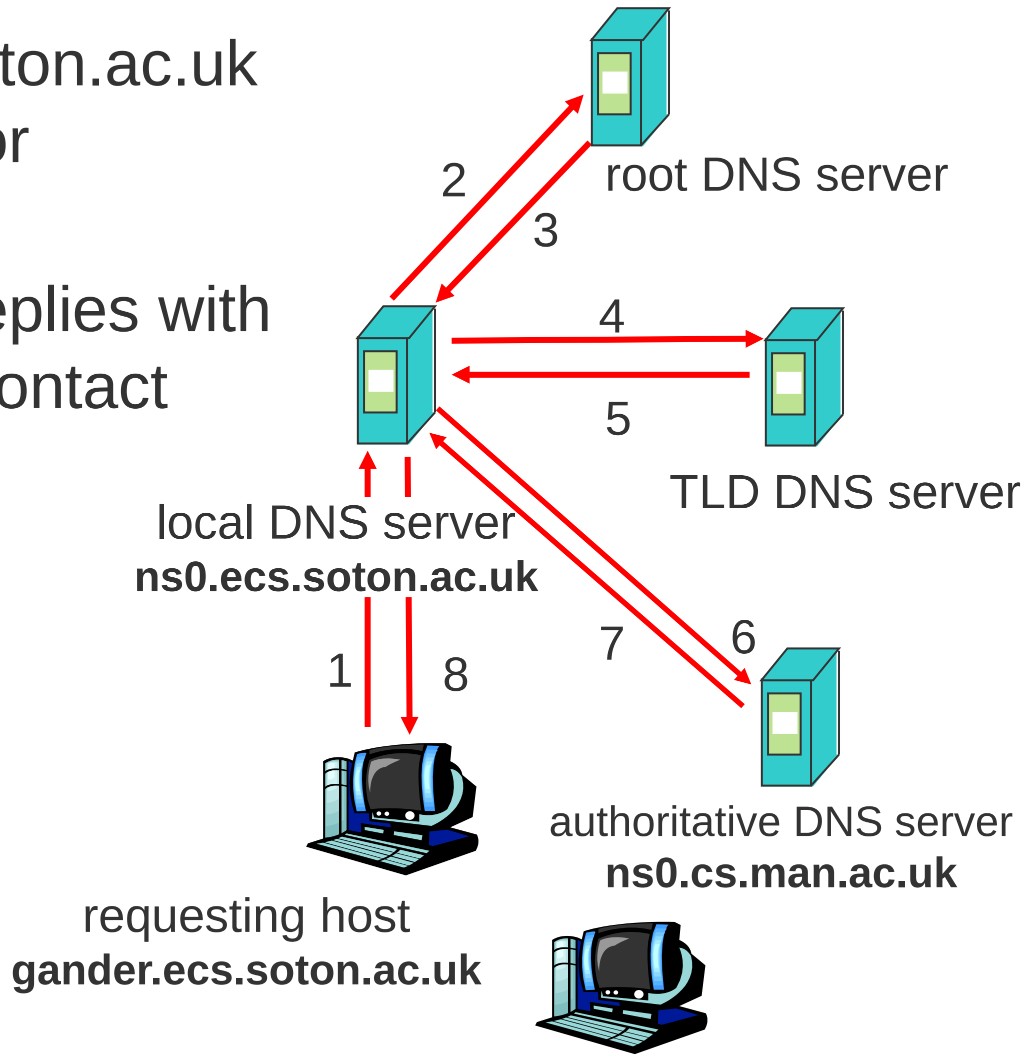
Auto configured


a Verisign, Dulles, VA
c Cogent, Herndon, VA (also LA)
d U Maryland College Park, MD
g US DoD Vienna, VA
h ARL Aberdeen, MD
j Verisign, (21 locations)
k RIPE London
(also 16 other locations)
i Autonomica, Stockholm
(plus 28 other locations)
e NASA Mt View, CA
f Internet Software C. Palo Alto, CA
(and 36 other locations)
m WIDE Tokyo
(also Seoul, Paris, SF)
b USC-ISI Marina del Rey, CA
l ICANN Los Angeles, CA

What happens an address when changes?

DNS: Iterative Resolution Example

- Host gander.ecs.soton.ac.uk wants IP address for ruby.cs.man.ac.uk
- Contacted server replies with name of server to contact



K&R:2.5.2, P&D:9.1.3

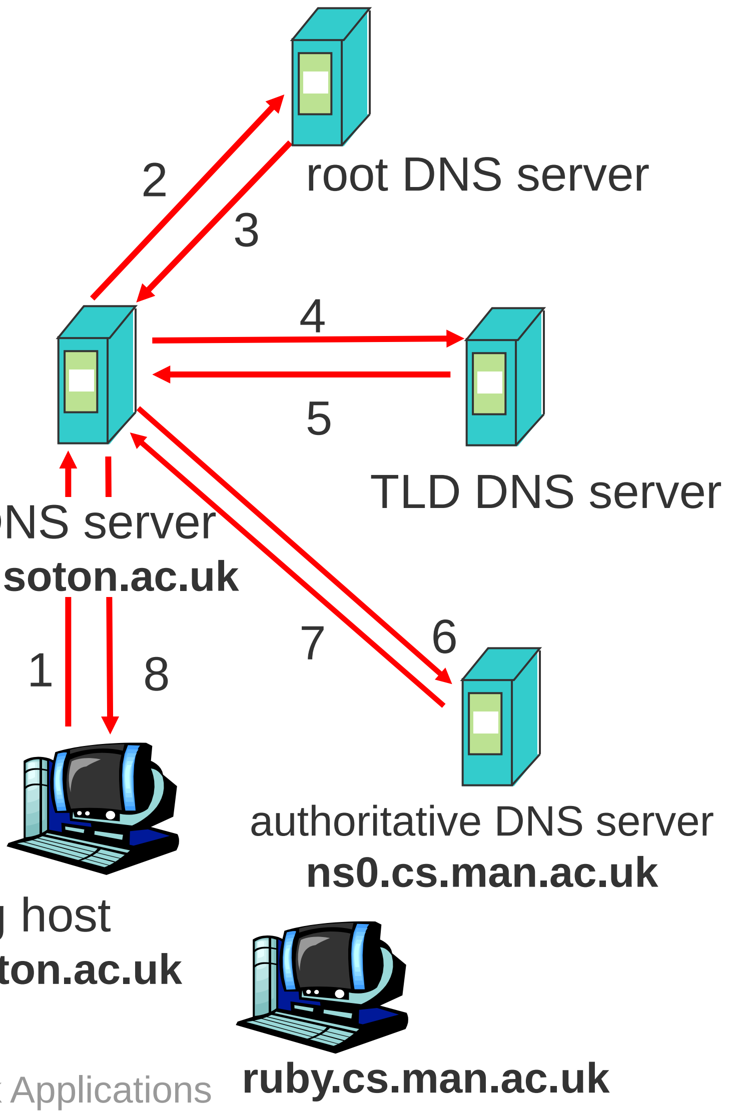
DNS: Iterative Resolution Example

- Host gander.ecs.soton.ac.uk wants IP address for ruby.cs.man.ac.uk
- Contacted server replies with name of server to contact

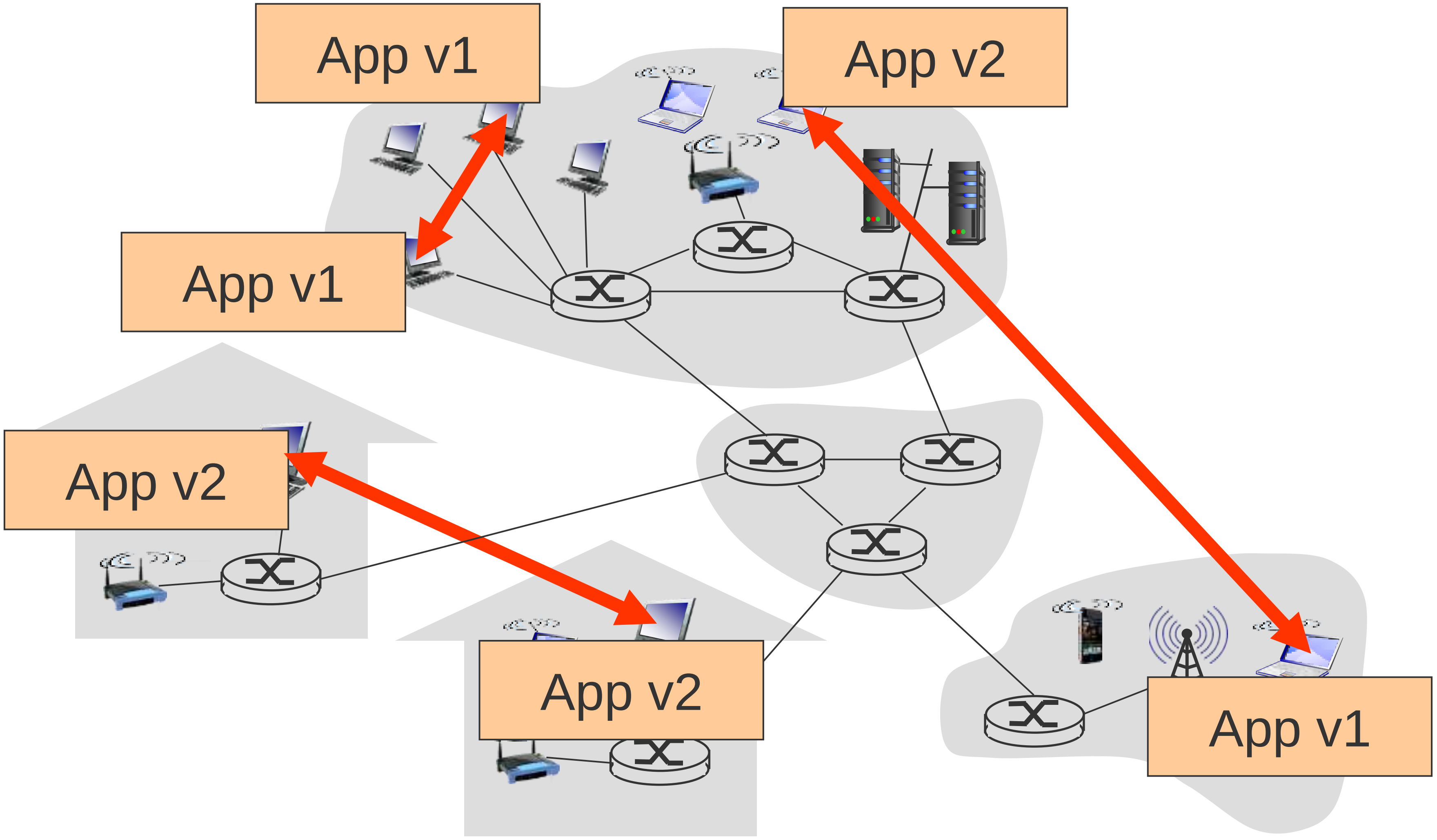
“I don’t know this name, but ask this server”

How many DNS messages?

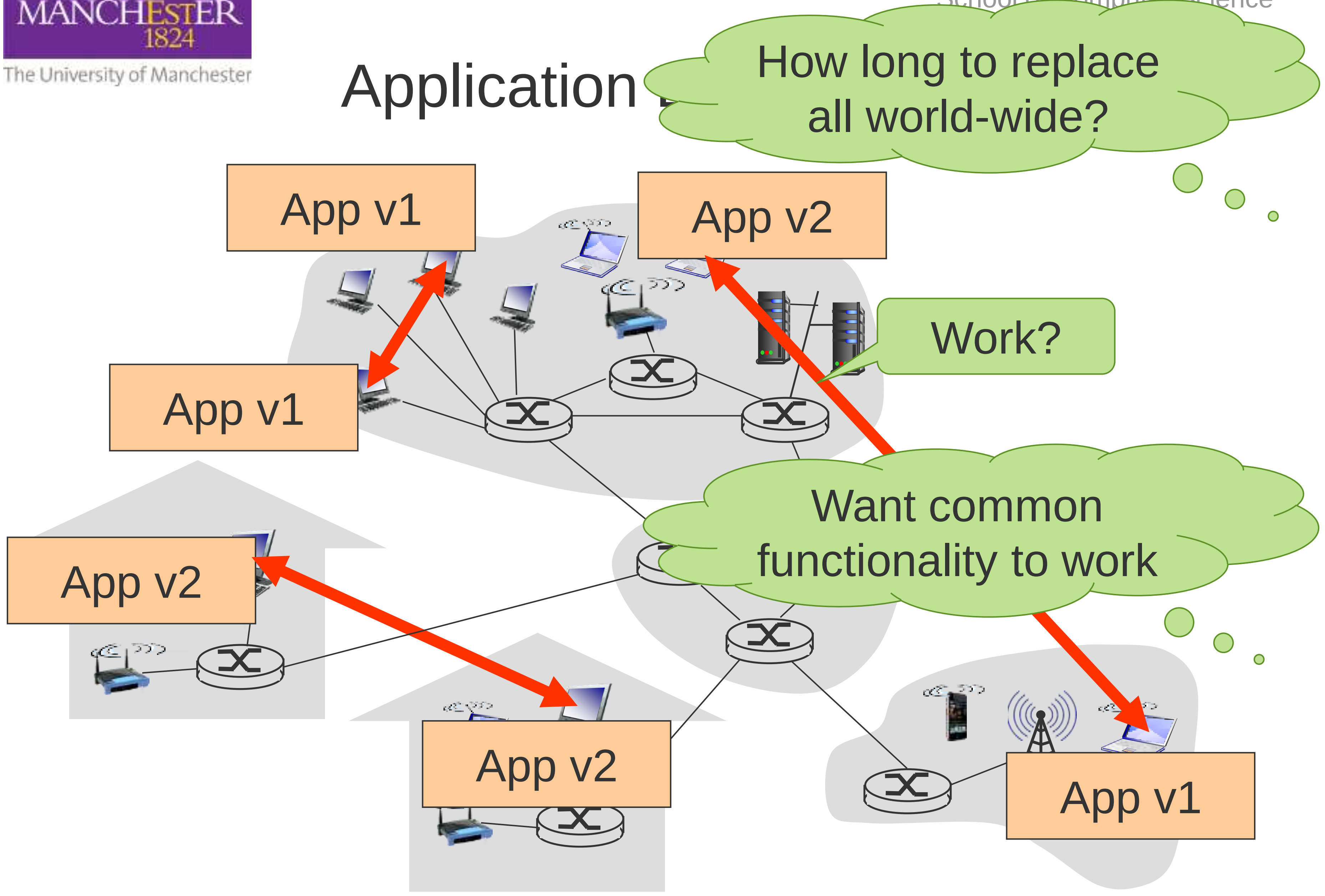
Local NS caches all



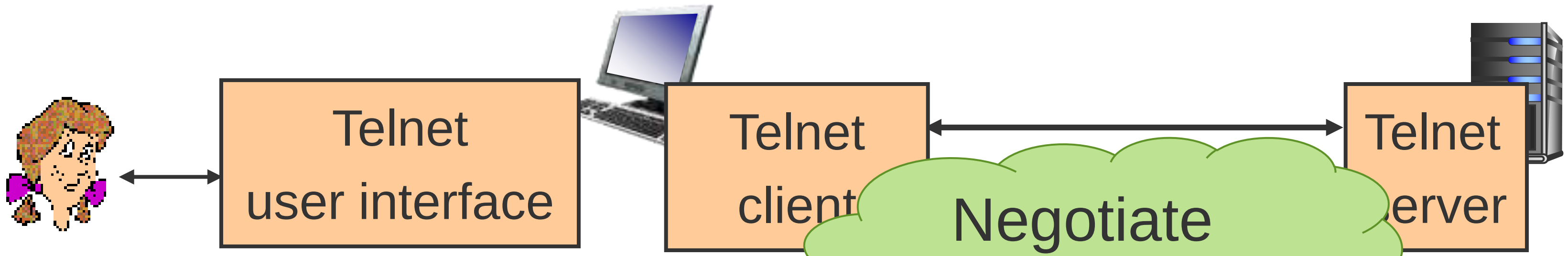
Application Extensibility



Application


How long to replace all world-wide?
App v1
App v2
App v1
Work?
Want common functionality to work
App v2
App v2
App v1

Extensibility: Telnet Options



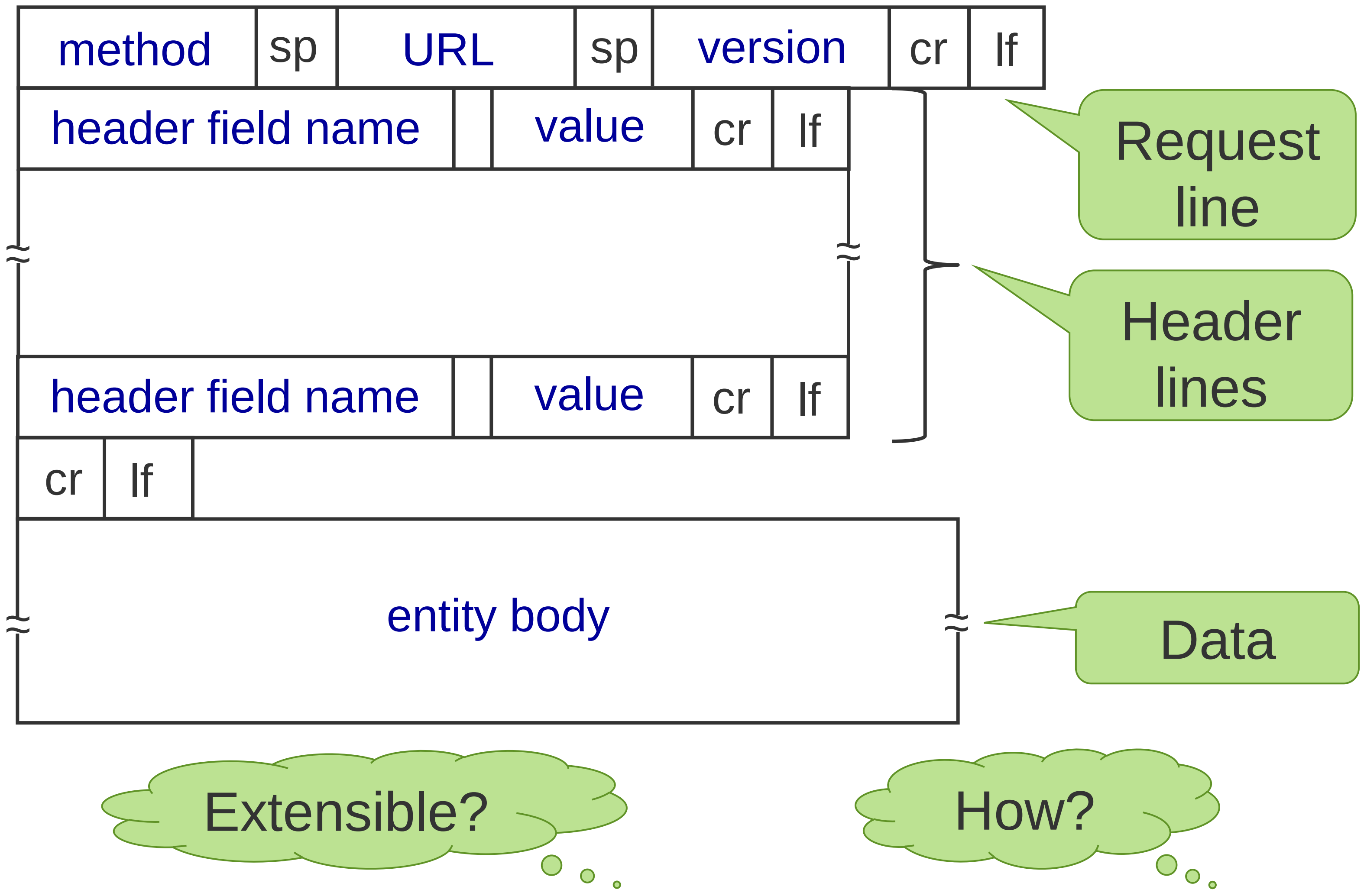
Negotiate options used

Send	Accept Response	Reject Response
DO	WILL	WON'T
DON'T	WON'T	WILL
WILL	DO	DON'T
WON'T	DON'T	DO

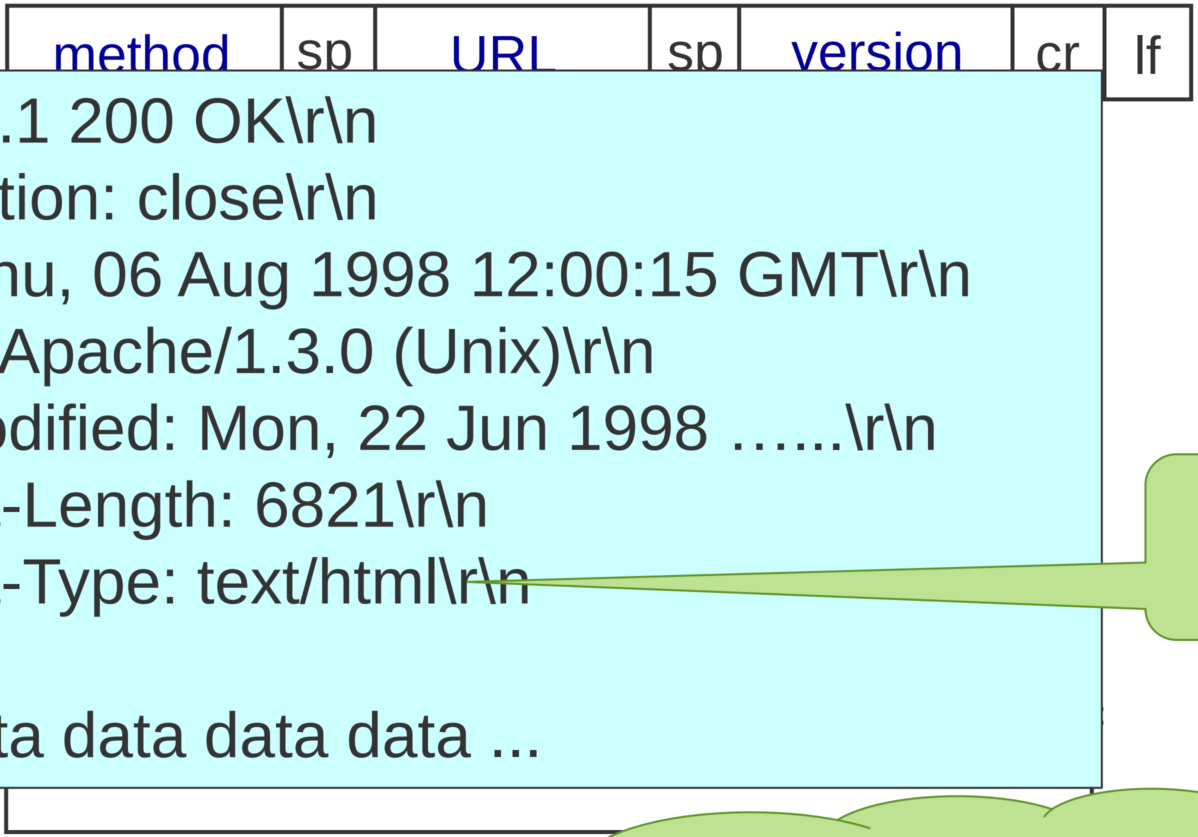
Standard only defines how to negotiate

Unknown requests are rejected

Extensibility: HTTP Messages



Extensibility: HTTP Messages

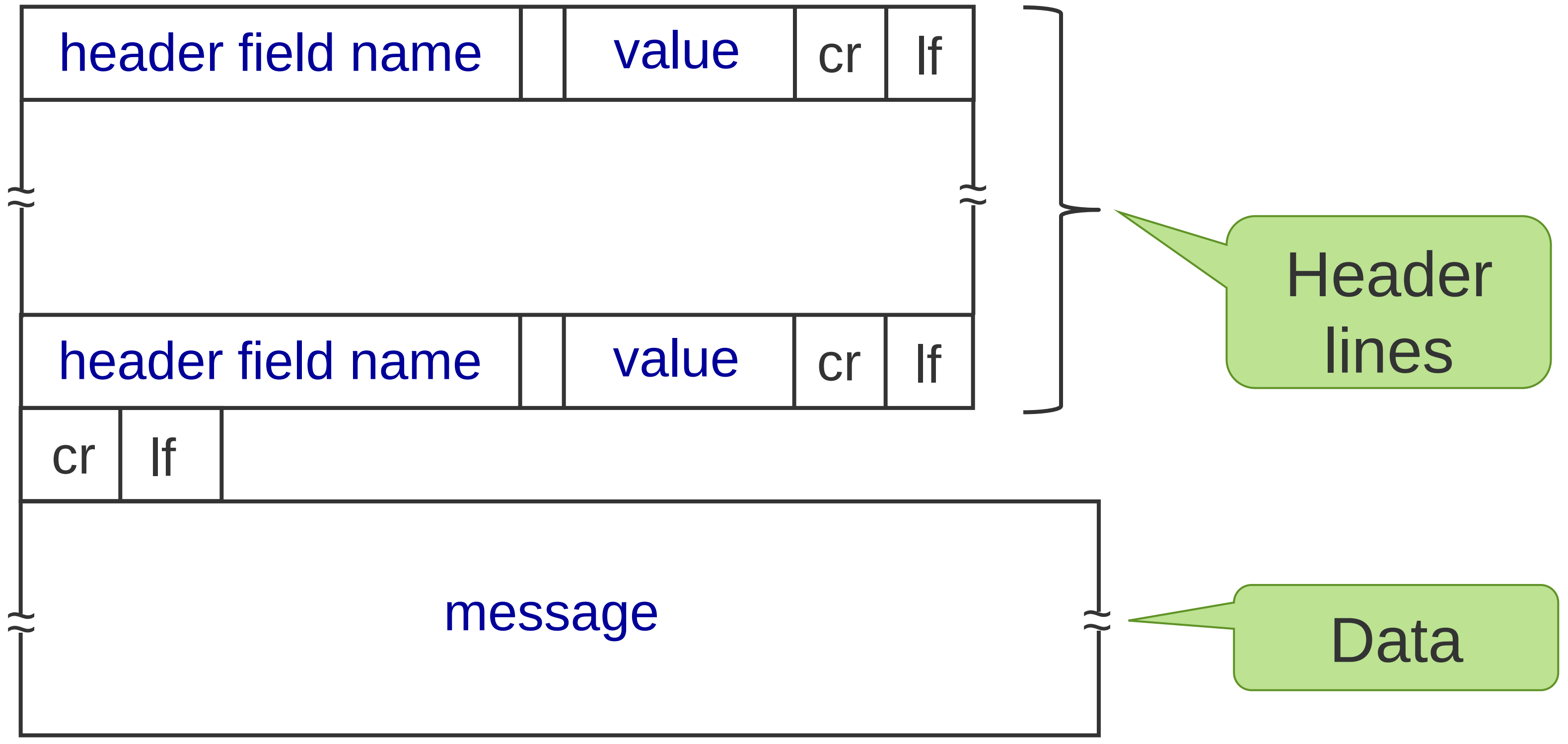


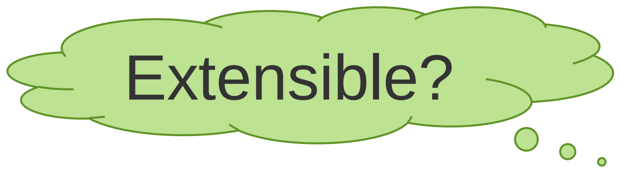
```
HTTP/1.1 200 OK\r\n
Connection: close\r\n
Date: Thu, 06 Aug 1998 12:00:15 GMT\r\n
Server: Apache/1.3.0 (Unix)\r\n
Last-Modified: Mon, 22 Jun 1998 ......\r\n
Content-Length: 6821\r\n
Content-Type: text/html\r\n
\r\n
data data data data data ...
```

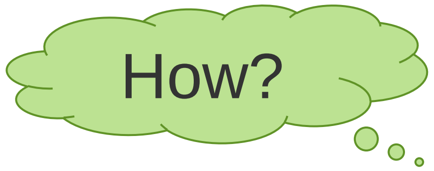
Indication of content type

Uses MIME definitions; see email

Extensibility: RFC822 Message







Extensibility: RFC822 Message

```
MIME-Version: 1.0\r\n
Content-Type: multipart/mixed: boundary="--xx"\r\n
From: Andy.Carpenter@cs.man.ac.uk\r\n
To: Second@cs.man.ac.uk\r\n
Subject: Useful information\r\n
Date: Mon, 07 Sep 1998 19:45:19\r\n
\r\n
--xx\r\n
Content-Type: text/plain; charset=us-ascii\r\n
Content-Transfer-Encoding: 7bit\r\n
\r\n
Here is the laboratory answer and a compiled
version
          Andy.
```

K&R:2.4.2, P&D:9.2.1

Email: MIME Messages

- Multipurpose Internet Mail Extensions; allow:
  - messages to use non-ASCII character sets
  - mail messages to carry different types of data
- Defines:
  - additional header lines
    - used to understand message
  - set of context types (and subtypes)
  - encodings to carry data
- What MIME messages can bring:
  - HTML format emails, attachments
- Defined in RFC2045/2046/2047/4288/4289/2049

Email: MIME Content Types

- Defined by Content-Type header field
- Discrete types; e.g.:
  - image/gif, image/jpeg, text/plain
- Application discrete types, subtype is application:
  - application/postscript, application/msword
- Message composite type
  - encapsulated message
- Multipart composite type
  - body contains several parts

Mapped to local function or application to display

Summary

- Good application design is good protocol design
- An application probably uses a collection of protocols
- Have content and data
  - request/response encapsulates data in control
  - control can be embedded in data (Telnet)
  - can separate control and data (FTP, RTP)
- Need to understand information transferred
- Extensible mechanisms
- Can reduce network traffic using caches
- Compared to lower levels, greater variety of requirements