Architecture of the World Wide Web
Web Information Systems

CS/INFO 431

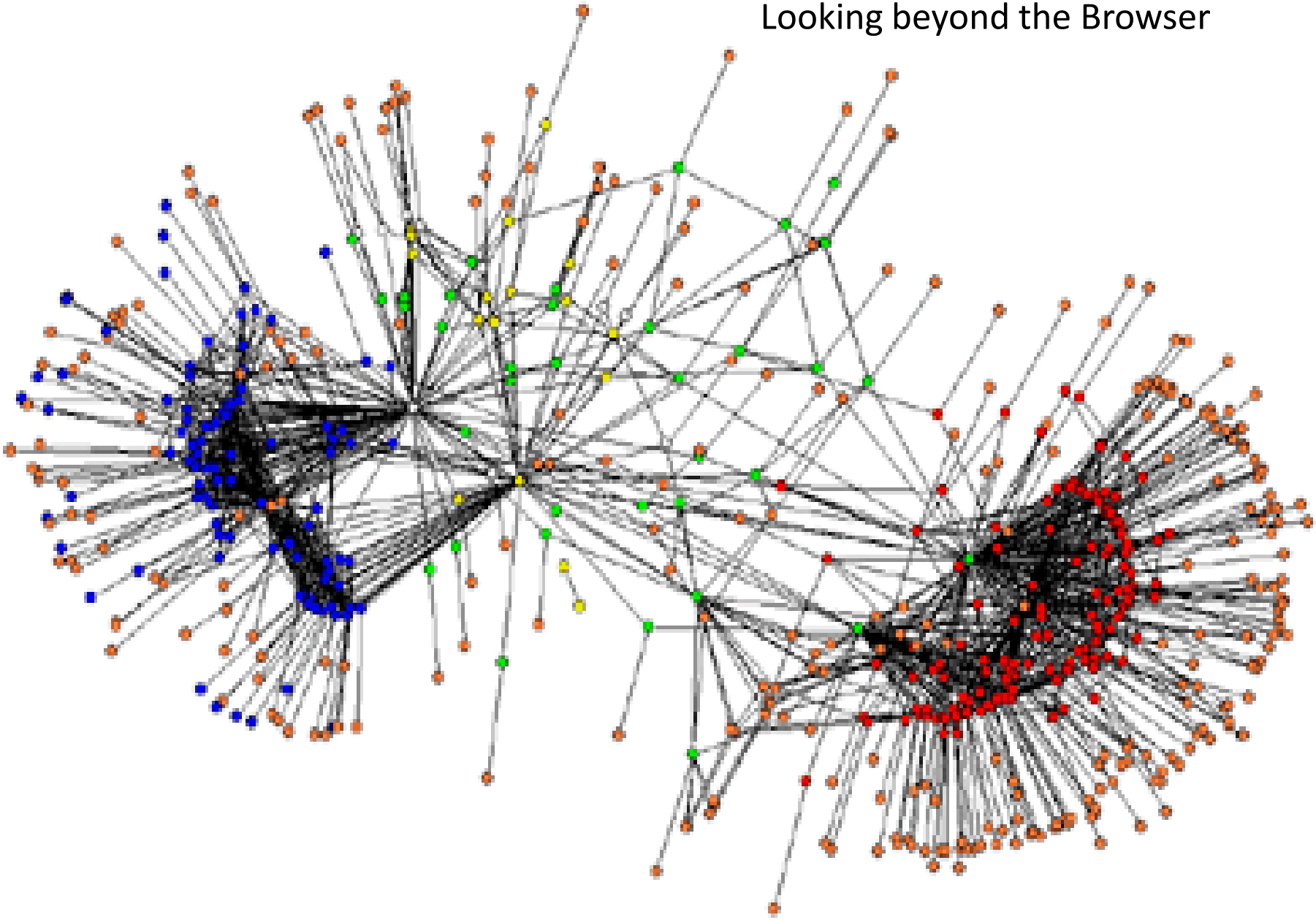
January 31, 2007
Carl Lagoze – Spring 2007

Acknowledgments

- Erik Wilde – UC Berkeley
  - http://dret.net/lectures/infosys-ws06/http

<?xml version="1.0" encoding="UTF-8"?>

Looking beyond the Browser

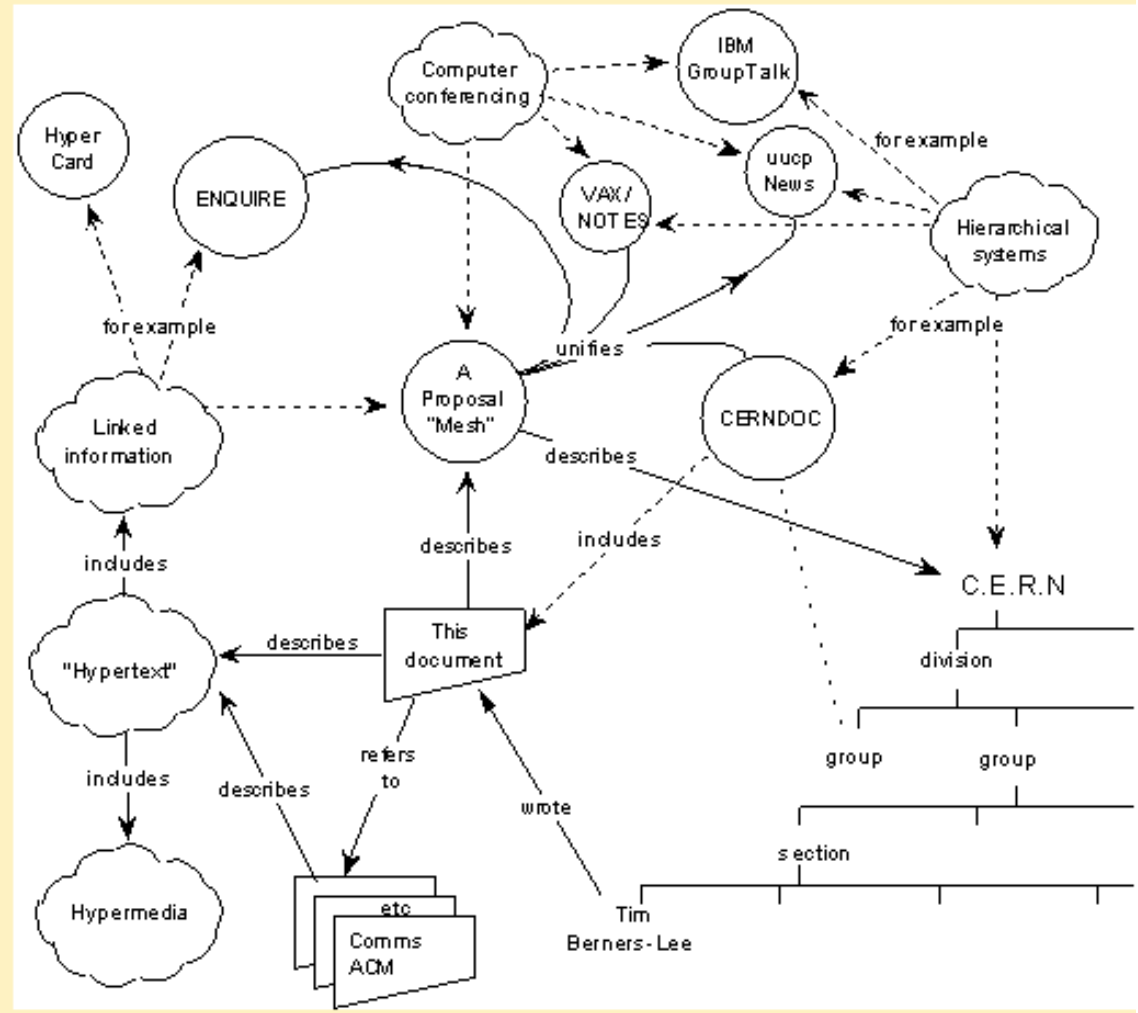


```
margin-left: 15px;
padding-bottom: 5px;
}
```

Information Management: A Proposal

Tim Berners-Lee, CERN
March 1989, May 1990

http://www.w3.org/History/1989/proposal.html



Web as a graph

We can call the circles nodes, and the arrows links. Suppose each node is like a small note, summary article, or comment. I'm not over concerned here with whether it has text or graphics or both. Ideally, it represents or describes one particular person or object. Examples of nodes can be

- People
- Software modules
- Groups of people
- Projects
- Concepts
- Documents
- Types of hardware
- Specific hardware objects

Web as a graph

The arrows which links circle A to circle B can mean, for example, that A...

- depends on B
- is part of B
- made B
- refers to B
- uses B
- is an example of B

Architecture of the World Wide Web, Volume One

W3C Recommendation 15 December 2004

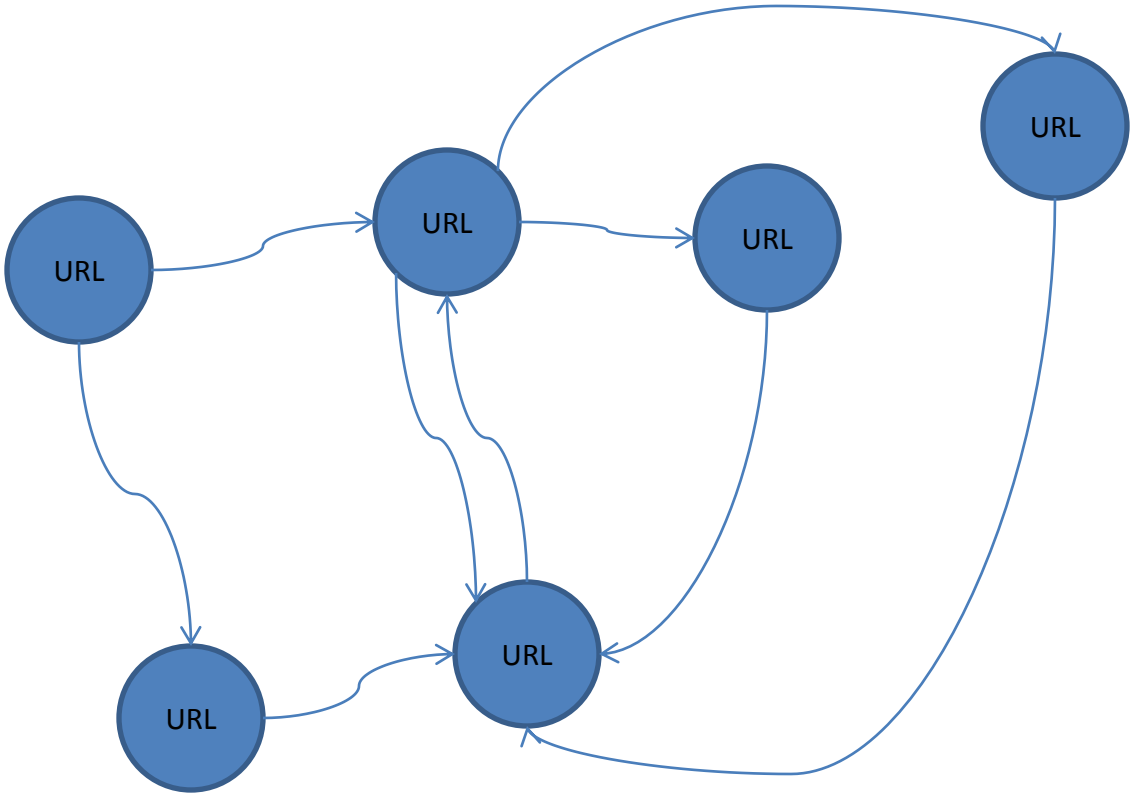
This version:
http://www.w3.org/TR/2004/REC-webarch-20041215/
Latest version:
http://www.w3.org/TR/webarch/
Previous version:
http://www.w3.org/TR/2004/PR-webarch-20041105/

Editors:
Ian Jacobs, W3C
Norman Walsh, Sun Microsystems, Inc.

Make Note: Three URLs for the "same" Intellectual object

http://www.w3.org/TR/webarch/

Naïve view of web graph



Think for a second…

- When I access google.com on my cell phone it looks different than on my desktop
- When I access google.com from Paris it looks different than when I access it from Ithaca

Architectural Components of the Web

- Participants
  - Servers
  - Web Agents
    - General agents such as robots – e.g., google crawler
    - specialized as User Agents
- Identification
  - URIs to identify Resources
    - Some URIs resolve URLs
    - Some do not resolve – INFO URIs
- Interaction
  - Standardized protocols with exchange of messages
    - One example HTTP
  - Requests result in return of Representations
- Formats
  - Representation is in form of sequence of bytes with a media type that provides hints to the agent about processing
  - One example of a format is a MIME type

A resource is…

- An entity that has an identity (a URI)
  - Some resources are digital – URI <-> URL
  - Some are non-digital – people, institutions, etc.
- Abstract – you can't examine/touch/see a resource
  - Information hiding
- A service point for initiating protocol (HTTP) actions
- Target of links (you make hyperlinks to a URI)
  - <a href="http://google.com">

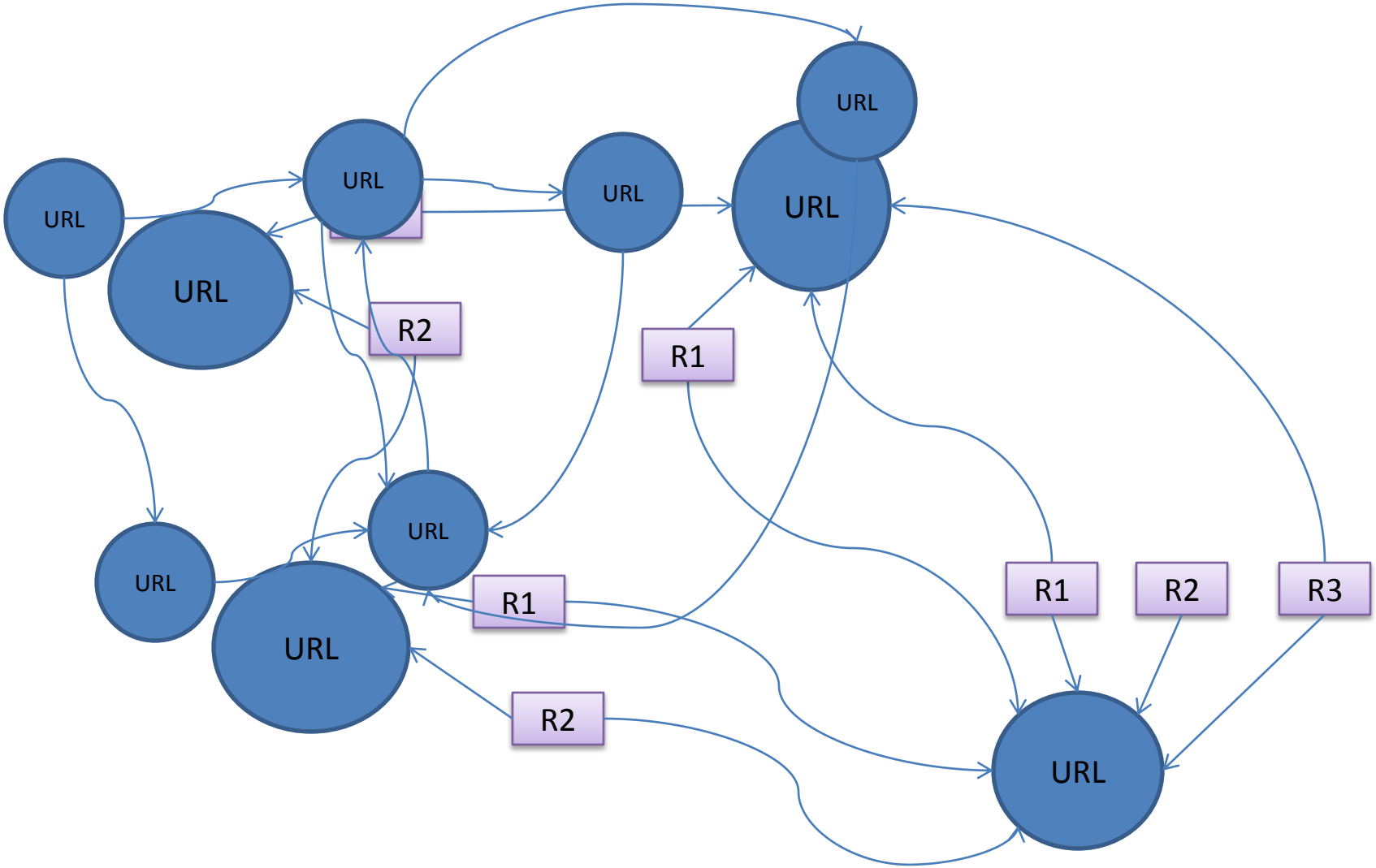
A representation is…

- The result of applying a service request upon a resource
- What the server determines to be the state of the resource
  - Parameterized
    - Time, space
    - Request parameters
  - Many to one mapping from resource to representation
- A package:
  - Metadata – about request, server actions, agent
  - Data – the "content"
- The entity that is processed by a web agent
  - In the case of a browser (user agent) rendered and displayed
  - Note that many agents such as crawlers make extensive use of metadata (last-modified)
- The entity that is the source of links
  - <a href="http://google.com">

Use cases of representations in increasing complexity

- Static transcription of a file
  - foo.html is disseminated from http://my.org/foo.html
- Dynamic dissemination of data based on static file
  - Translation from base jpeg to thumbnail
  - Result of PHP program
- Multiple representations from resource based on user agent or other parameters
  - google for cell phones and desktops
  - Content negotiation
- Totally time dependent representations
  - http://cnn.com

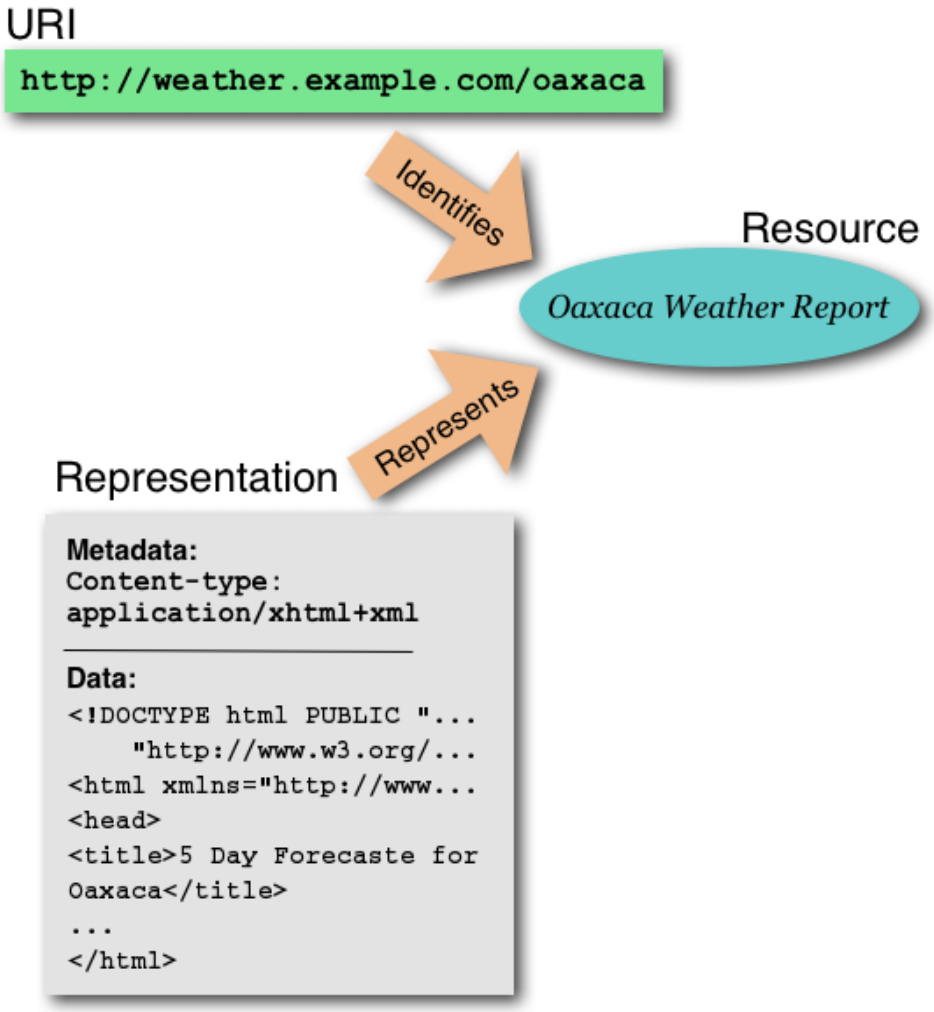
Real nature of web graph



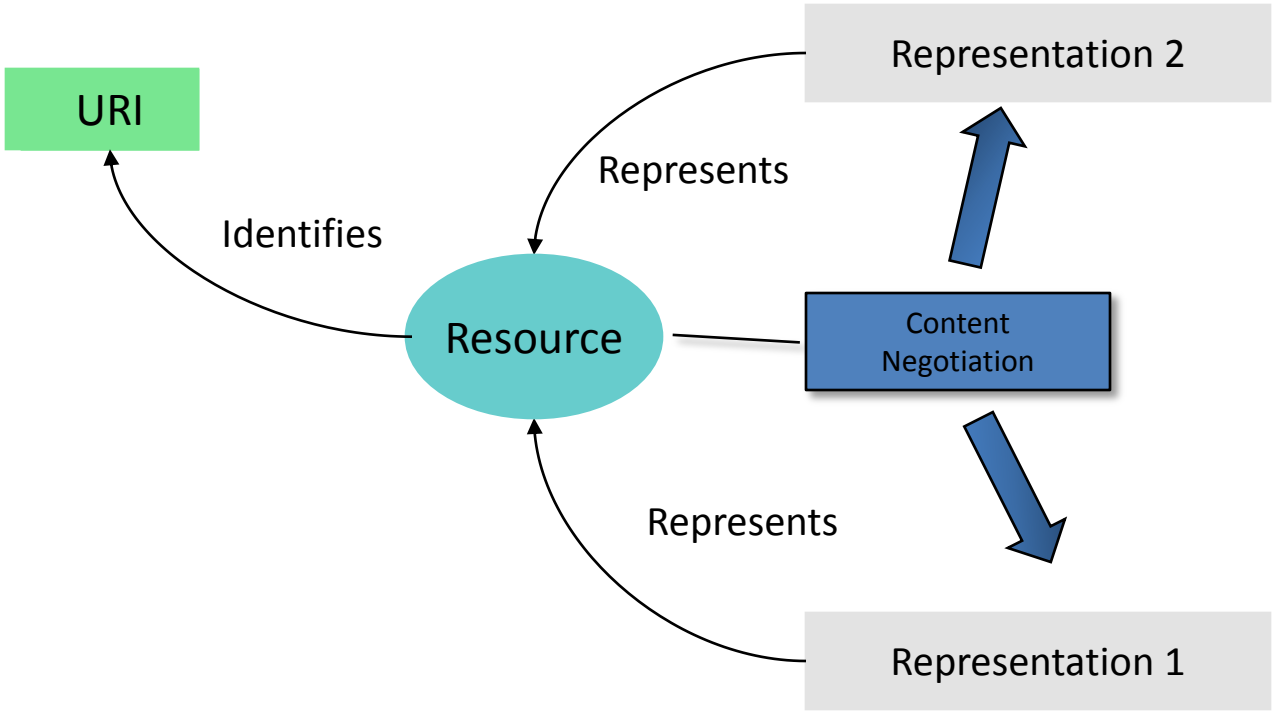
Think about it…

The web graph is completely ephemeral: Based on representations that are time and agent context

Closer look at Resource/Representation Relationship



Content Negotiation

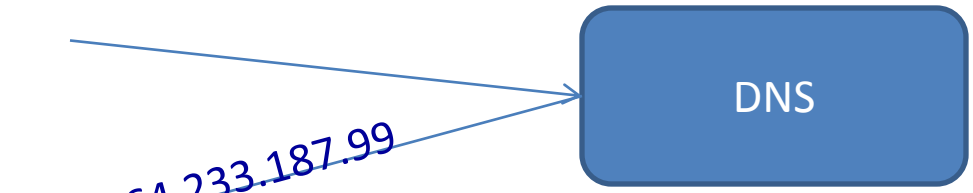


HTTP (Hypertext Transfer Protocol)

- HTTP 1.1 – RFC 2516 ftp://ftp.isi.edu/in-notes/rfc2616.txt
- Basis of interaction between web agents and servers
- Layered on top of TCP
- Uses DNS
- Text-based
  - All messages and requests are human readable
- Stateless
  - No persistent client/server connection
  - All state carried in protocol request/reply (cookies)

Simple HTTP interaction

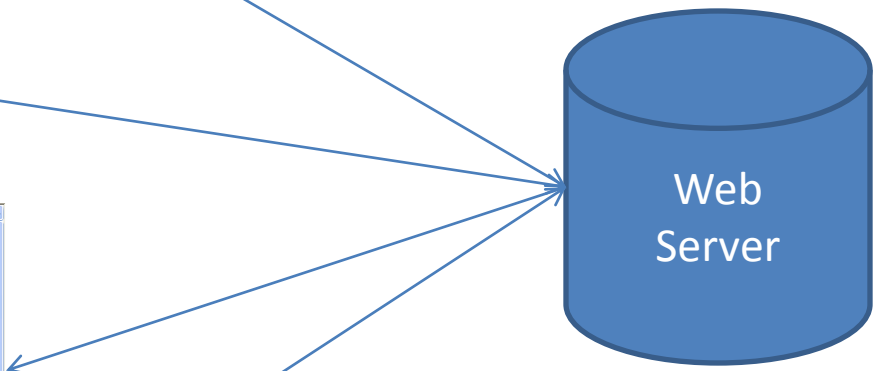
http://google.com

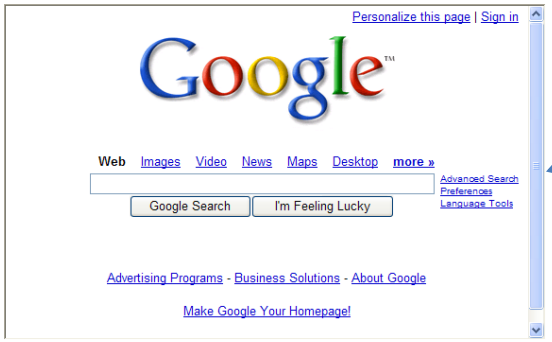


64.233.187.99

TCP open circuit

HTTP GET
index.html





TCP close circuit

HTTP Request Types

- GET – initiate a retrieval action based on the URI
- POST – transmit information package to URI at server
- HEAD – same as GET but return only header (metadata) information
- PUT – store the request content at the URI
- DELETE – Remove resource at URI

HTTP Example

```
-bash-2.05b$ telnet google.com 80
Trying 72.14.207.99...
Connected to google.com (72.14.207.99).
Escape character is '^]'.
GET index.html HTTP/1.1
Host: lagoze.com

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Set-Cookie: PREF=ID=c9f0e0565db57456:TM=1138635078:LM=1138635078:S=H-BsvXLg58YkL
l14; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Server: GWS/2.1
Transfer-Encoding: chunked
Date: Mon, 30 Jan 2006 15:31:18 GMT

bd8
<html><head><meta http-equiv="content-type" content="text/html; charset=ISO-8859
-1"><title>Google</title><style><!--
body,td,a,p,.h{font-family:arial,sans-serif;}
.h{font-size: 20px;}
.q{color:#0000cc;}
//-->
</style>
<script>
<!--
function sf(){document.f.q.focus();}
// -->
</script>
</head><body bgcolor=#ffffff text=#000000 link=#0000cc vlink=#551a8b alink=#ff00
00 onLoad=sf() topmargin=3 marginheight=3><center><table border=0 cellspacing=0
cellpadding=0 width=100%><tr><td align=right nowrap><font size=-1><a href="/url?
sa=p&pref=ig&pval=2&q=http://www.index.html/ig%3Fhl%3Den">Personalized Home</a><
/font></td></tr><tr height=4><td><img alt="" width=1 height=1></td></tr></table>
<table cellpadding=0 cellspacing=0 border=0><tr><td align=right valign=bottom><i
```

TCP Connection

HTTP Request

HTTP Response Headers (metadata)

HTTP Response Data (chunked)

HTTP Request

- Start line
  - Consists of method, path, version

    `GET index.html HTTP/1.1`
  - Valid methods include:
    - GET, POST, HEAD, PUT, DELETE
- Headers
  - HTTP/1.1 requires a Host: header

    `Host: cs.cornell.edu`
  - Many other headers
- Optional body content

HTTP Response

- Start line
  - consists of HTTP version, status code, and description

```
HTTP/1.1 200 OK
HTTP/1.1 404 Not Found
```

- Headers (Many header definitions)

```
Content-type: text/html
```

- Content

HTTP Response Codes

- Response coded by first digit
  - 1xx: informational, request received
  - 2xx: success, request accepted
  - 3xx: redirection
  - 4xx: client error
  - 5xx: server error

Simple HTTP GET - Response

GET /path/file.html HTTP/1.1
Host: cs.cornell.edu
User-Agent: Mozilla/3.0
[Blank line]

HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 31 Jan 2007 14:58:57 GMT
Content-Length: 1354

<html>
<head>
…

Simple HTTP Post

POST /path/script.php HTTP/1.1
Host: cs.cornell.edu
User-Agent: Mozilla/3.0
Content-Type: application/x-www-form-urlencoded
 Content-Length: 32

home=Cosby&favorite+flavor=flies
[Blank line]

HTTP Content Negotiation (server-side)

- Resources may have multiple dimensions
- General idea
  - Web agent makes HTTP requests stating constraints
  - Using constraints server decides on “best” representation
- HTTP defined constraints are language, encoding, format, character encoding
  - Accept, Accept-Charset, Accept-Encoding, Accept-Language
- Server may also use:
  - User-agent: device specificity
  - Host: localization
  - Cookies

Header request examples for content negotiation

```
Accept-Language: fr; q=1.0, en; q=0.5
Accept: text/html; q=1.0, text/*; q=0.8, image/gif; q=0.6,
      image/jpeg; q=0.6, image/*; q=0.5, */*; q=0.1
```

Other types of content negotiation

- Client-side
  - Server response with list of different representations
  - Client (or user) makes a choice
- Transparent
  - Cache plays a role in client-side negotiation