UNIVERSITY OF
Southampton

UNIVERSITY OF Southampton

The Architecture of the World Wide Web

COMP3220 Web Infrastructure

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

The evolving architecture of the Web

In its earliest days, the Web was defined primarily in terms of its implementation

- HTTP was what Web browsers and Web servers spoke
- HTML was what Web browsers read

As the Web grew, this became an increasing problem

- Network issues with early HTTP affected scalability
- Nature of application interactions on the Web changed (images, etc)
- Limited support for shared caching

The Web architecture addresses this by answering the following questions:

- How do all the Web components fit together in a consistent manner?
- How should future components fit together with the rest of the Web?

To properly understand the Web architecture,
we need to understand what the Web is

To properly understand what the Web is,
we need to understand how the Web came to be

What is the Web?

The Web is a distributed information system that provides access to hypertext documents and other objects of interest

We have a general name for these objects of interest:

resources

What is a resource?

The earliest formal uses of the term "resource" in an Internet context date from 1994

- Synonymous with "file" or "network object"
- Earliest reference to resources in IETF RFCs is in RFC1580 from March 1994
- First formal definition of Uniform Resource Identifiers in RFC1630 from June 1994
- Earlier documents submitted by Berners-Lee to IETF talked about documents

EARN Staff (1994) *Guide to Network Resource Tools.* RFC1580. Available online at https://tools.ietf.org/html/rfc1580
Berners-Lee, T. (1994) *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web.* RFC1630. Available online at https://tools.ietf.org/html/rfc1630

What is a resource?

“Familiar examples [of resources] include an electronic document, an image, a source of information with a consistent purpose (e.g., ‘today's weather report for Los Angeles’), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., ‘parent’ or ‘employee’), or numeric values (e.g., zero, one, and infinity).”

Berners-Lee, T. et al (2005) Uniform Resource Identifier (URI): Generic Syntax. RFC3986. Available online at https://tools.ietf.org/html/rfc3986

What is a resource?

"Familiar examples [of resources] include an electronic document, an image, a source of information with a consistent purpose (e.g., 'today's weather report for Los Angeles'), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., 'parent' or 'employee'), or numeric values (e.g., zero, one, and infinity)."

A resource may change over time yet still be considered the same resource

- The Ship of Theseus/my grandfather's axe/Trigger's broom

Berners-Lee, T. et al (2005) Uniform Resource Identifier (URI): Generic Syntax. RFC3986. Available online at https://tools.ietf.org/html/rfc3986

What is a resource?

"Familiar examples [of resources] include an electronic document, an image, a source of information with a consistent purpose (e.g., 'today's weather report for Los Angeles'), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., 'parent' or 'employee'), or numeric values (e.g., zero, one, and infinity)."

A resource may not be something which can be "retrieved"

- Can still be interacted with over the network

Berners-Lee, T. et al (2005) Uniform Resource Identifier (URI): Generic Syntax. RFC3986. Available online at https://tools.ietf.org/html/rfc3986

What is a resource?

"Familiar examples [of resources] include an electronic document, an image, a source of information with a consistent purpose (e.g., 'today's weather report for Los Angeles'), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., 'parent' or 'employee'), or numeric values (e.g., zero, one, and infinity)."

A resource may not even be something that's "on the Internet"

Berners-Lee, T. et al (2005) Uniform Resource Identifier (URI): Generic Syntax. RFC3986. Available online at https://tools.ietf.org/html/rfc3986

Resource

Architectural Bases of the Web

The notion of a resource is central to the architecture of the Web

We need to be able to:

- *identify* them
- *represent* them
- *interact* with them

These are orthogonal considerations

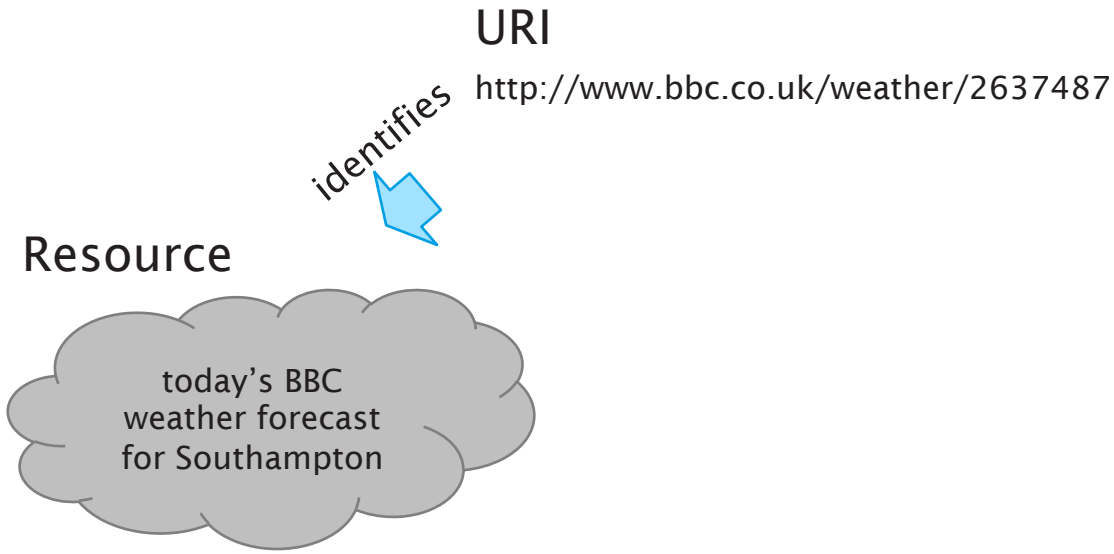
Identification

We need a way to identify resources

- Machines need to be able to resolve those identifiers (to "fetch" the resource)
- Humans should (ideally) be able to read and write those identifiers

Uniform Resource Identifiers

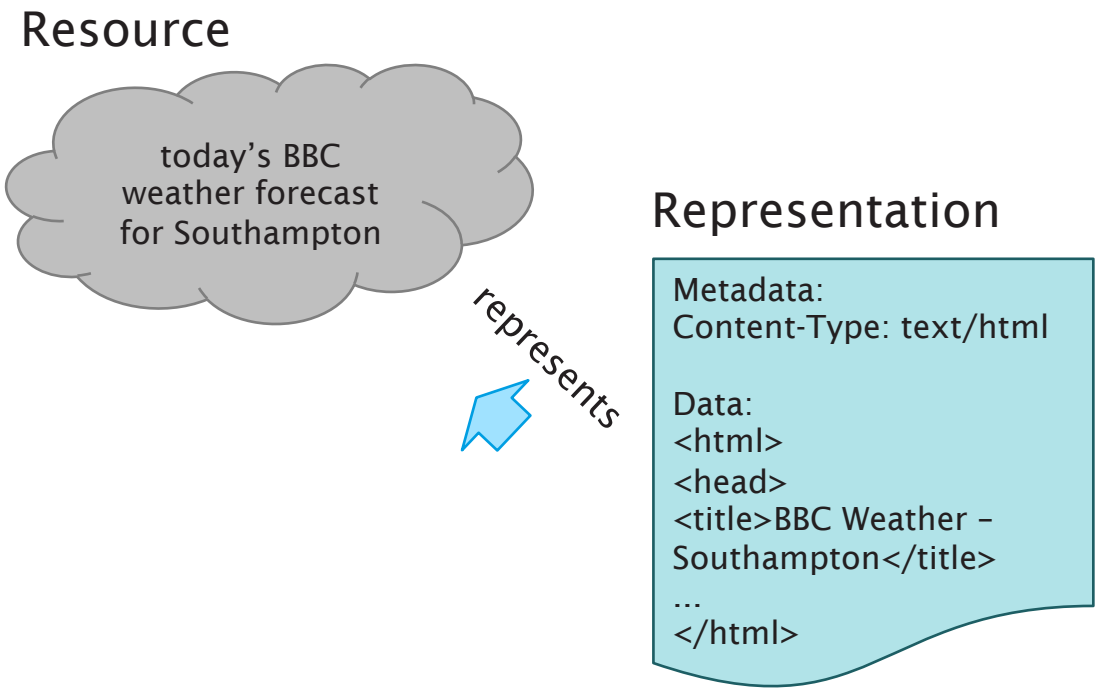
- Arguably the single most important contribution of the Web



Representation

A representation is data that encodes information about resource state.

Representations do not necessarily describe the resource, or portray a likeness of the resource, or represent the resource in other senses of the word "represent".

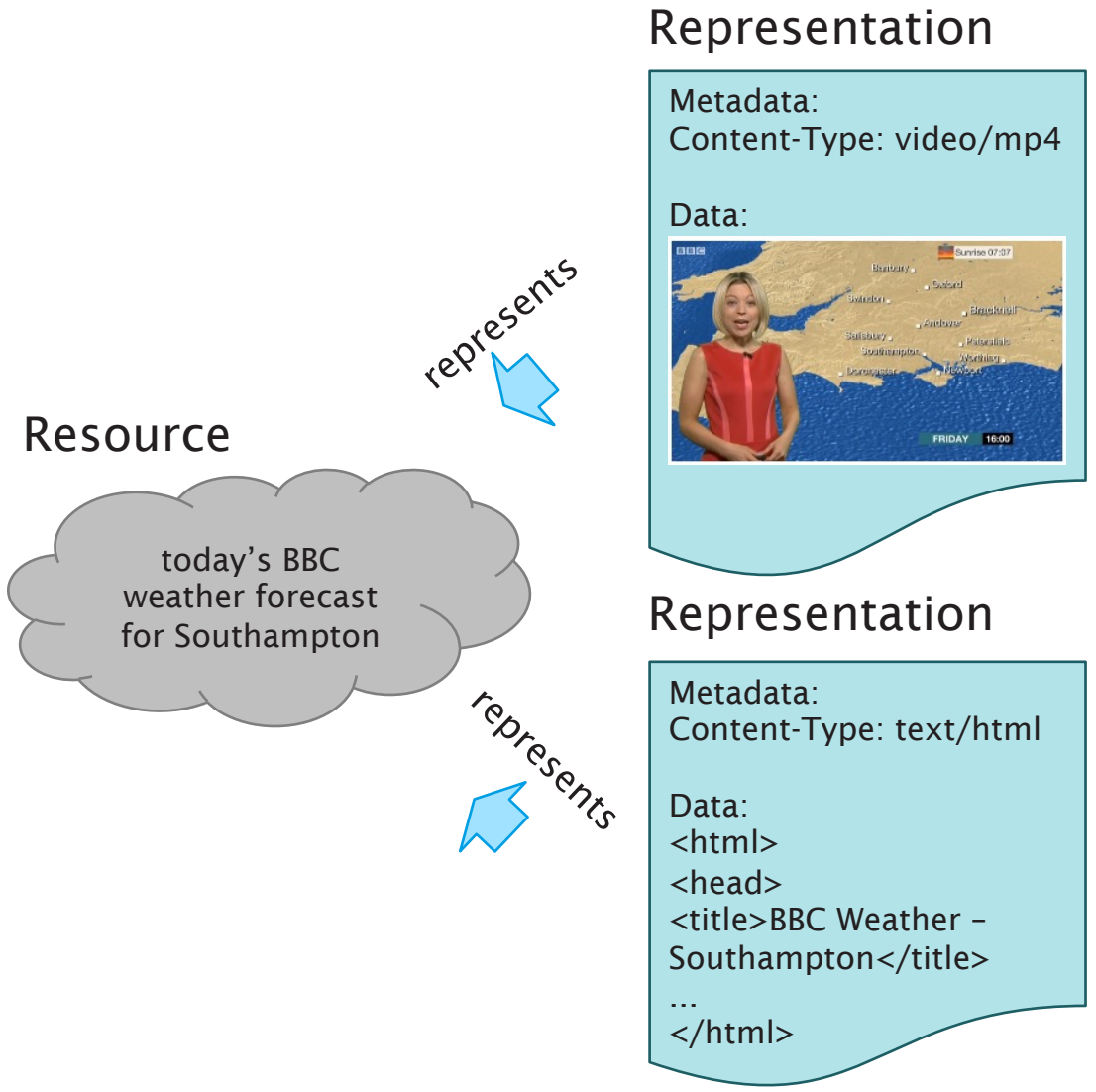


Representation

A representation is data that encodes information about resource state

Representations do not necessarily describe the resource, or portray a likeness of the resource, or represent the resource in other senses of the word "represent"

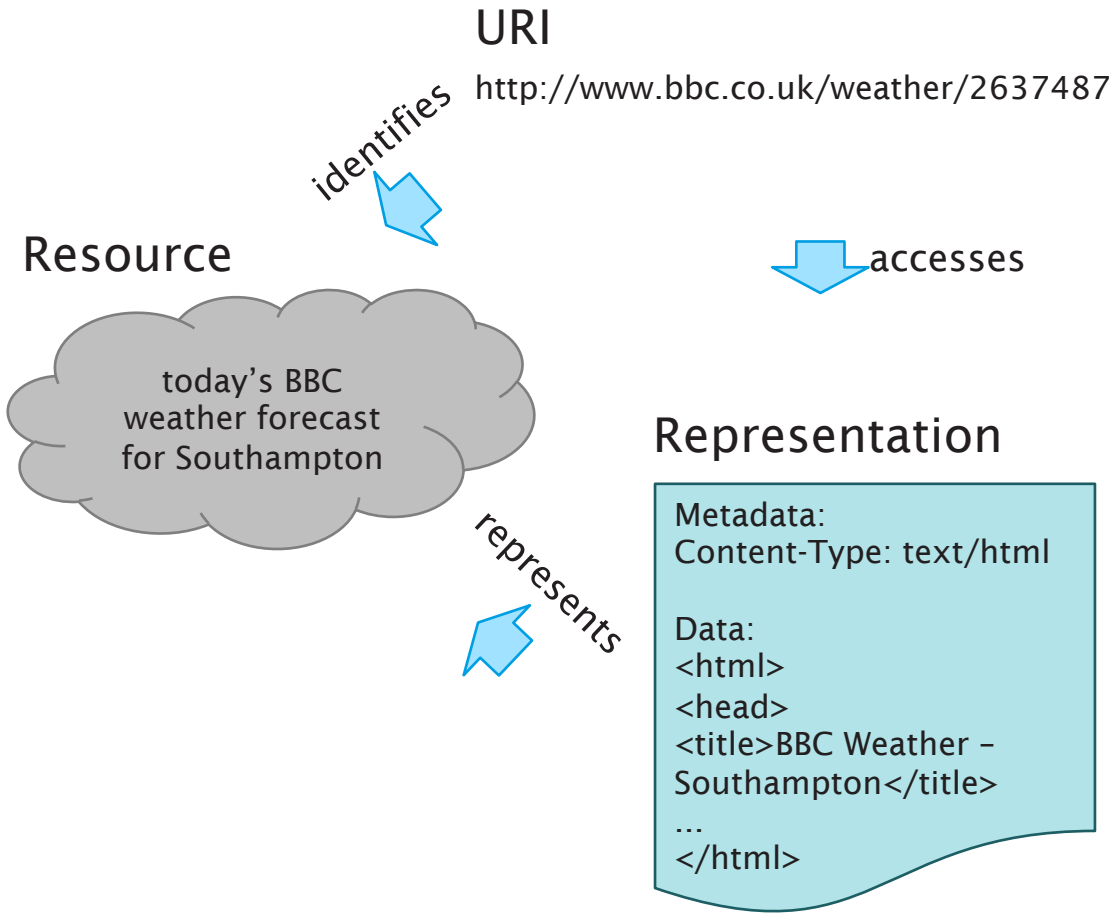
May be multiple representations of a given resource



Interaction

Resource representations are transmitted using interaction protocols.

The interactions between Web agents and resources are defined in terms of protocols that specify the exchange of messages



Representational State Transfer

The core Web architecture was heavily informed by the work of Roy Fielding

- Co-author of HTTP and URI RFCs
- Co-founder of the Apache Software Foundation

Fielding defined an architectural style known as *representational state transfer* or *REST*

- What data elements exist?
- What components (i.e. processing elements) exist?
- What constraints apply to the interactions between components?

Fielding, R.T. (2000) *Architectural Styles and the Design of Network-based Software Architectures*. PhD Thesis. University of California at Irvine. Available online at http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Data Elements

Resources

Identifiers (URIs, etc)

Representations (HTML, etc)

Metadata (resource, representation, control)

Components

Origin servers

- Definitive source of resource representations
- Web server: Apache, nginx, IIS, etc

Gateways

- Intermediary selected by origin server
- Often used to integrate legacy services

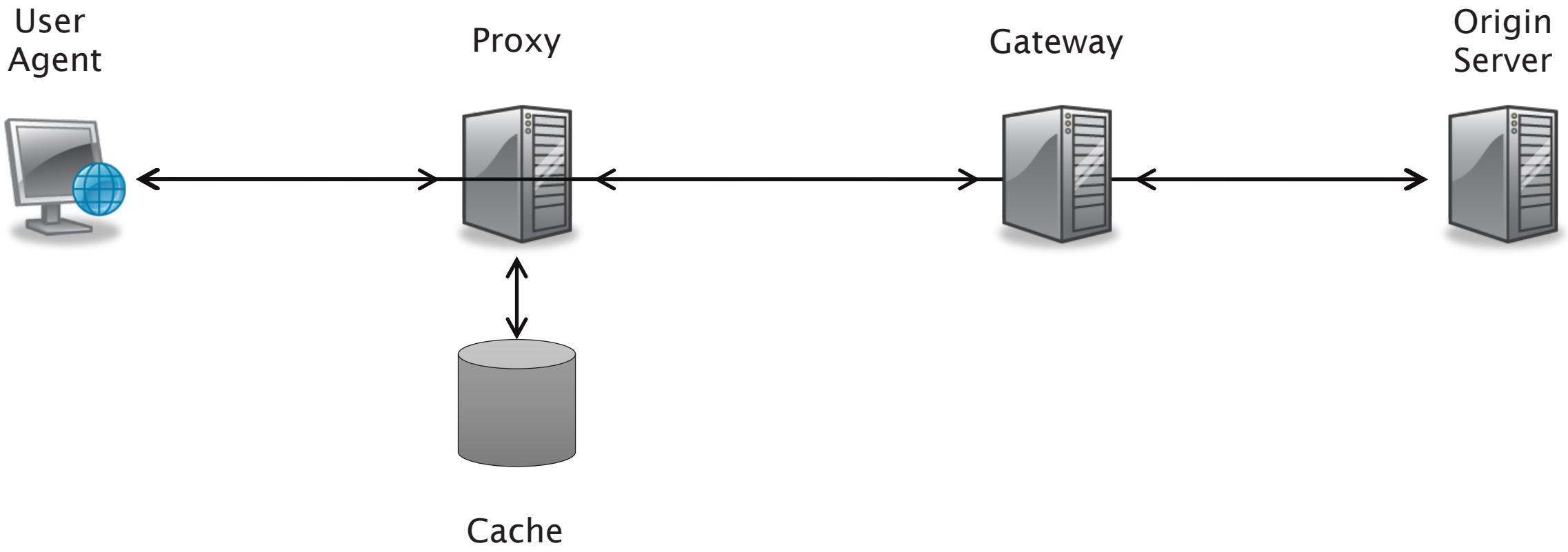
Proxies

- Intermediary selected by user agent
- Filter, cache, etc

User agents

- Browser: Chrome, IE, Safari, Firefox, etc

Components



Constraints

- Client-Server

Separation of concerns:

- user interface (client)
- data storage (server)

+ Improves portability of user interface
+ Improves scalability by simplifying server
+ Allows components to evolve separately

Constraints

- Client-Server
- Stateless

Each request from client to server must contain all of the information necessary to understand the request

- No context is stored on the server
- Session state is kept entirely on the client

+ Improves visibility
  (can consider each request in isolation)
+ Improves reliability
  (easier to recover from partial failures)
+ Improves scalability
  (reduces server resource usage)
– Increases per-interaction overhead

Constraints

- Client-Server
- Stateless
- Caching

Response data must be labelled as cacheable or non-cacheable

- If cacheable, client may reuse response data for later requests

+ Allows some interactions to be eliminated
+ Reduces average latency of interactions
– Stale data reduces reliability

Constraints

- Client-Server
- Stateless
- Caching
- Uniform Interface

Uniform interface between components

- Identification of resources
- Manipulation of resources through representations
- Self-descriptive messages
- Hypermedia as the engine of application state (HATEOAS)

+Improves visibility of interactions

+Encourages independent evolvability

–Degrades efficiency
(depending on optimisation)

Constraints

- Client-Server
- Stateless
- Caching
- Uniform Interface
- Layered

System components have no knowledge of components beyond those with which they directly interact

- Encapsulate legacy services
- Introduce intermediaries

+Limits system complexity

+Improves scalability (load balancing)

−Adds latency and overhead (offset by caching)

Constraints

- Client-Server
- Stateless
- Caching
- Uniform Interface
- Layered
- Code on Demand (optional)

Client functionality extended by downloading and executing code

- Applets
- Scripts

+Improves extensibility

−Reduces visibility

Further Reading

Jacobs, I. and Walsh, N. (2004) *Architecture of the World Wide Web, Volume One*. W3C Recommendation.

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

Fielding, R.T. (2000) *Architectural Styles and the Design of Network-based Software Architectures*. PhD Thesis. University of California at Irvine. Chapters 4-5.

http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

UNIVERSITY OF Southampton

Next Lecture: Identification, Representation, Interaction