REST and Web Services

Instructor: Yongjie Zheng

CS 441: Software Engineering

World Wide Web

- **Uniform Resource Identifier (URI)**: a string of characters used to identify a name or a resource.
- **Hypertext Transfer Protocol (HTTP)**: an *application protocol* for distributed hypermedia systems.
- **Hypertext Markup Language (HTML)**: the language for displaying web pages. A HTML document consists of tags and plain texts.

About HTTP

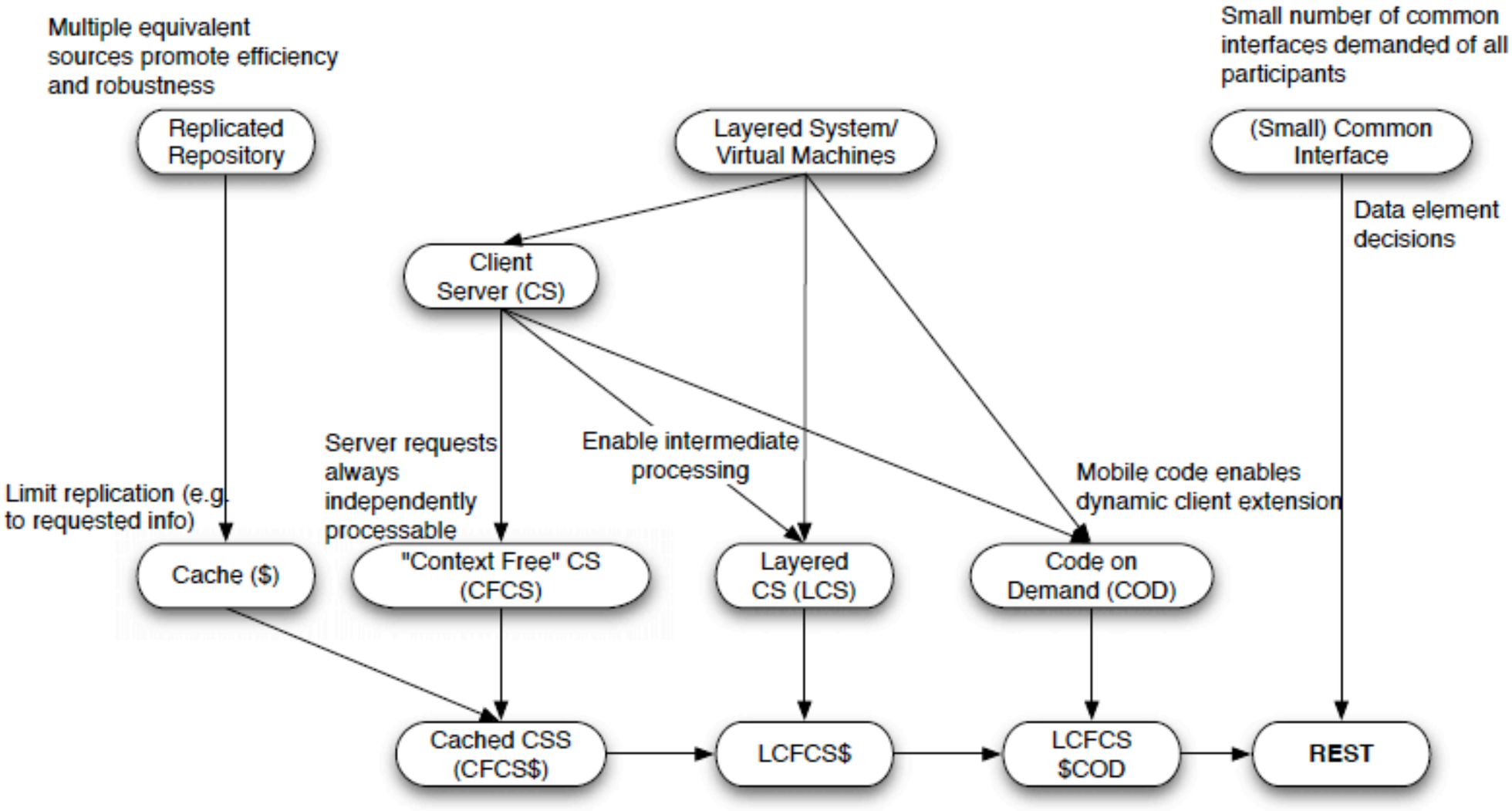
- HTTP methods
  - GET: retrieve the information.
  - POST: submit the information.
  - PUT: store and replace the information.
  - DELETE: delete the information.
- HTTP follows the REST architecture style

Representational **S**tate **T**ransfer (REST)

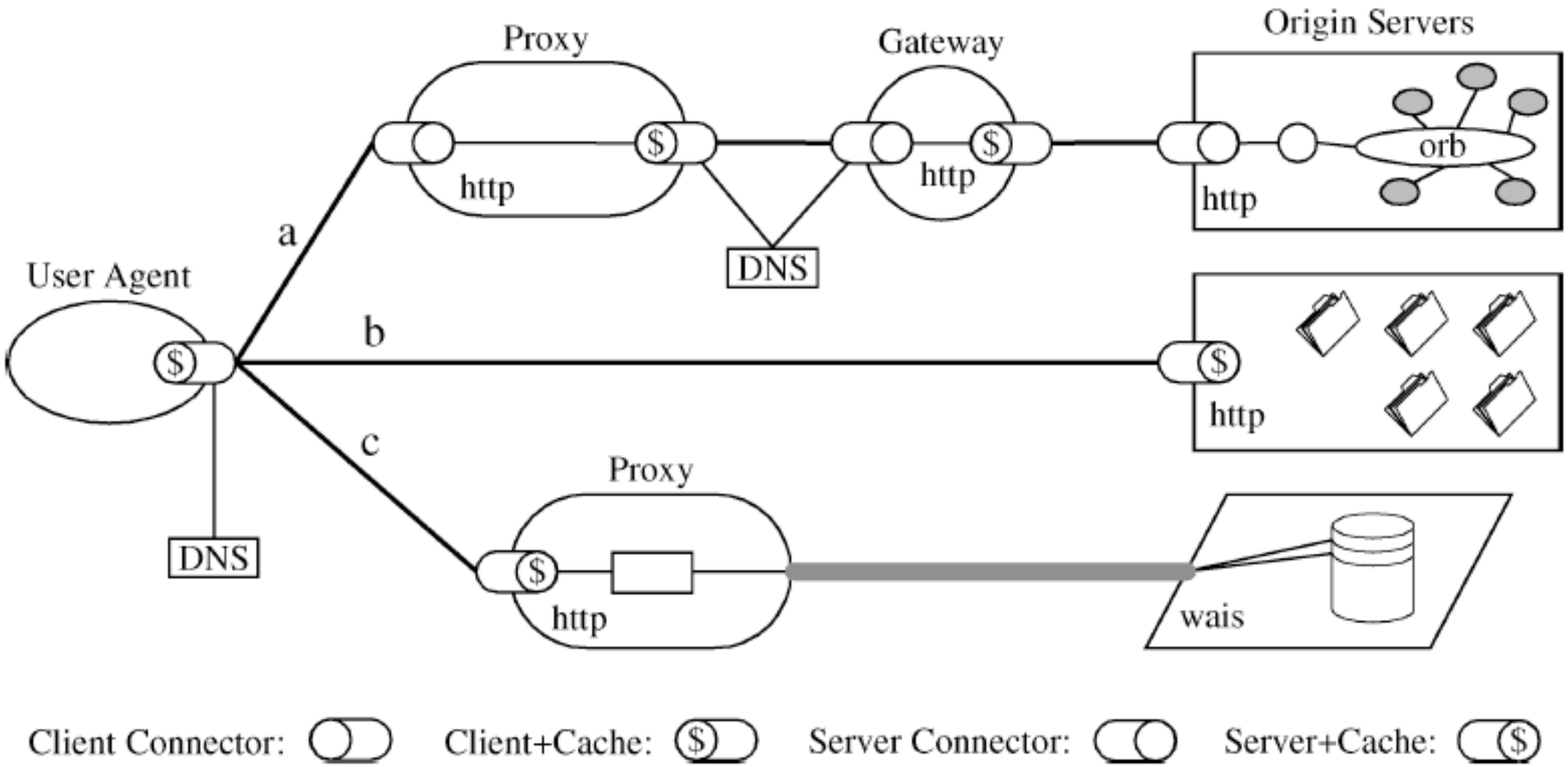
- REST is an architecture style of a distributed hypermedia system - the Web.
- Defined and evolved by Roy Fielding from 1994 to 2000, when he was a Ph.D. student at UC Irvine.
- About the name: "... to evoke how a well-designed Web application behaves: a network of Web pages forms **a virtual state machine**, allowing a user to progress through the application by selecting a link or submitting a short data-entry form, with each action resulting in a transition to the next state of the application by **transferring a representation of that state** to the user." - [Fielding & Taylor - 2002]

Derivation of REST

1. Client-Server (C-S)
2. Stateless (C-S-S)
3. Cache (C-$-S-S) -- the architecture style of the old Web (before 1994).
4. Uniform Interface
5. Layered
6. Code-on-demand



Derivation of REST: a hybrid architecture style.



An Instance of REST

REST Principles

- [RP1] *The key abstraction of information is a resource,named by an URL.* Any information that can be named can be a resource.
- [RP2] *The representation of a resource is a sequence of bytes, plus representation metadata to describe those bytes.* The particular form of the representation can be negotiated between REST components.

REST Principles (cont'd)

- [RP3] *All interactions are context-free*. Each interaction contains all of the information necessary to understand the request, independent of any requests that may have preceded it.
- [RP4] *Only a few primitive operations are available*. Components perform only a small set of well defined methods on a resource producing a representation to capture the current or intended state of that resource and transfer that representation between components. These methods are global to the specific architectural instantiation of REST; for instance, all resources exposed via HTTP are expected to support each operation identically.

REST Principles (cont'd)

- [RP5] *Idempotent operations and representation metadata are encouraged in support of caching and representation reuse.*
- [RP6] *The presence of intermediaries is promoted.* Filtering or redirection intermediaries may also use both the metadata and the representations within requests or responses to augment, restrict, or modify requests and responses in a manner that is transparent to both the user agent and the origin server.

Mismatches of REST

- Cookies
  - Using cookies means not all application state is carried in the message.
  - Cookies break visibility: caches do not understand them.
    - Cookies break the Back button.
- AJAX (e.g. Google Maps): "REST's goal was to reduce server-side state load; in turn, AJAX reduces server-side computational load. ... AJAX expands our notion of resources." - [Erenkrantz &Gorlick&Taylor]
- Web Services

Web Services

- **Definition**: A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. [W3C]

Web Services

- Standards
  - SOAP (Simple Object Access Protocol)
  - WSDL (Web Service Description Language)
  - UDDI (Universal Description, Discovery, and Integration)
- Advantages
  - Platform independent
  - Language independent
  - Machine-to-machine communication

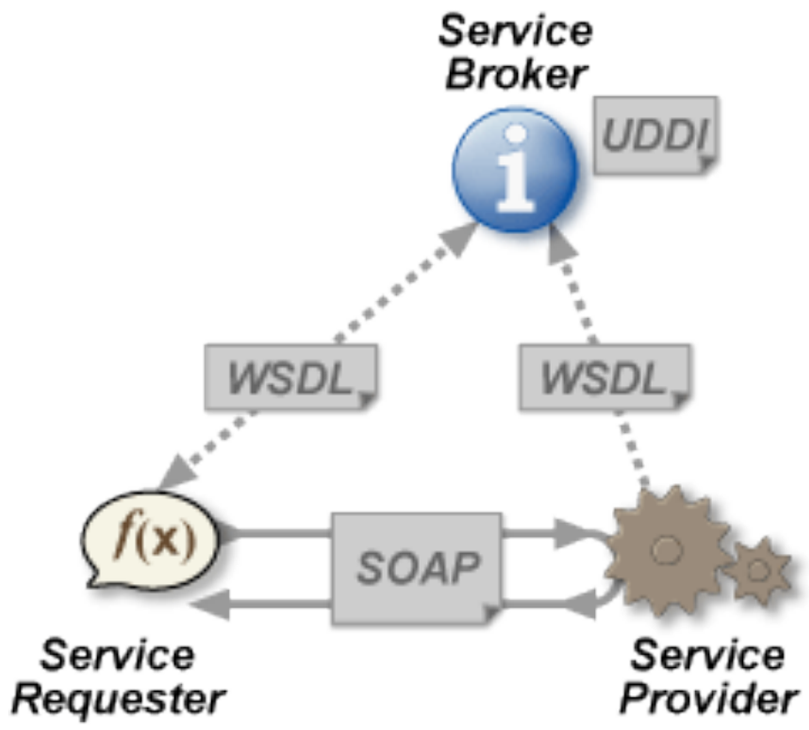


Diagram from [Wiki]

Web Services

- **RESTful, resource-oriented web services**: the service uses HTTP method (GET, PUT, POST, DELETE) for the method information, and exposes a URI for every piece of data that the client may want to operate on.
- **RPC-style (traditional) web services**: the service exposes only one URI (the "endpoint"), and supports only one method on that URI (POST). It ignores most HTTP features.
- **REST-RPC hybrid web services**: the service embeds the method information in URI, and exposes multiple URIs.

```
GET /upc?value=001441000055 HTTP/1.1
Host: www.upcdatabase.com
...
```

An example of RESTful web service request.

```
GET /upc?method=lookupUPC&value=001441000055 HTTP/1.1
Host: www.upcdatabase.com
...
```

An example of hybrid-style web service request.

```
POST /rpc HTTP/1.1
Host: www.upcdatabase.com
User-Agent: XMLRPC::Client (Ruby 1.8.4)
Content-Type: text/xml; charset=utf-8
Content-Length: 158
Connection: keep-alive

<?xml version="1.0" ?>
 <methodCall>
  <methodName>lookupUPC</methodName>
 <params>
  <param><value><string>001441000055</string></value></param>
 </params>
</methodCall>
```

An example of SOAP Request over HTTP (Non-RESTful RPC-Style Web Services).

Web Services

	Storage of method information	Storage of scoping information	Examples
RESTful, resource-oriented	HTTP Method	URI	Static websites, Amazon's S3 web services
RPC-style	SOAP Envelop	SOAP Envelop	Traditional SOAP-based web services
Hybrid	URI	URI	Web applications created without understanding REST

More about Web Services

- RESTful web services
  - HTTP is an application protocol; it doesn't send bits, it transfers representational state.
  - If the body of a POST or PUT is not a piece of representational state, you're not doing REST.
  - The list of HTTP methods is fixed for all resources.

More about Web Services

- Non-RESTful web services
  - Treat HTTP as a transport protocol like TCP. In essence, they are building new protocols (SOAP) and tunneling them over existing application protocols (HTTP).
  - Instead of addressing resources, Non-RESTful web services address software components.
  - POST, POST, POST!

References

- Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. Dissertation. 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

- Roy Fielding and Richard N. Taylor. Principled design of the Modern Web Architecture. ACM Transactions on Internet Technology, 2, 2, pp. 115-150 (May 2002).

- Richardson, Leonard, and Sam Ruby. RESTful web services. O'Reilly Media, Incorporated, 2007.