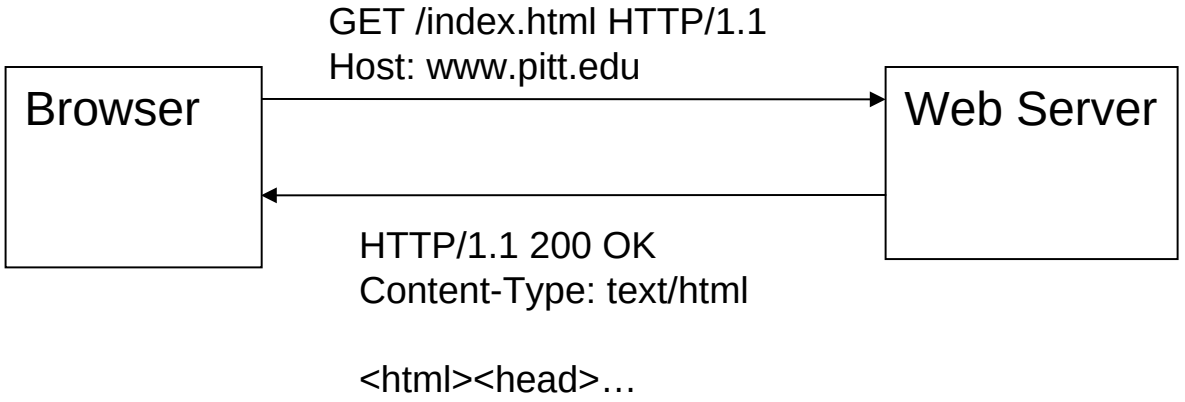
Representational State Transfer (REST): Representing Information in Web 2.0 Applications

Hypertext Transfer Protocol (HTTP)

- A communications protocol
- Allows retrieving inter-linked text documents (hypertext)
  - World Wide Web.
- HTTP Verbs
  - HEAD
  - **GET**
  - **POST**
  - PUT
  - DELETE
  - TRACE
  - OPTIONS
  - CONNECT



Representational State Transfer (REST)

- A style of software architecture for distributed hypermedia systems such as the World Wide Web.
- Introduced in the doctoral dissertation of Roy Fielding
  - One of the principal authors of the HTTP specification.
- A collection of network architecture principles which outline how resources are defined and addressed

REST and HTTP

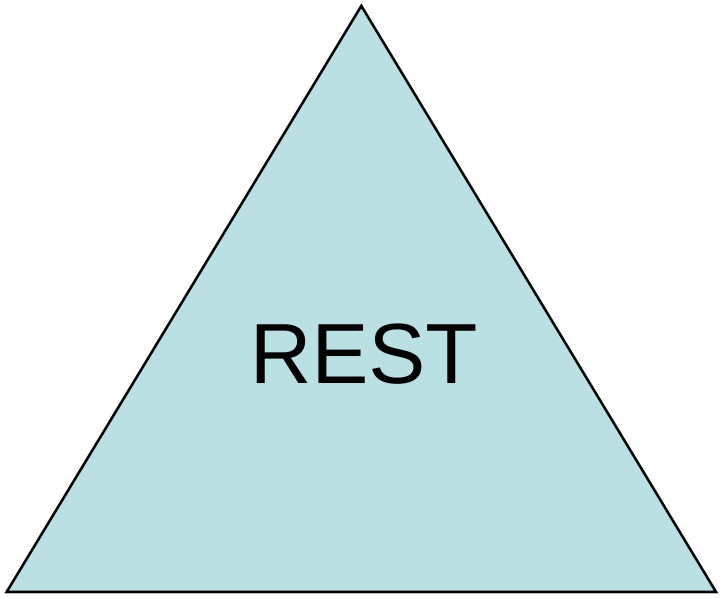
- The motivation for REST was to capture the characteristics of the Web which made the Web successful.
  - URI Addressable resources
  - HTTP Protocol
  - Make a Request – Receive Response – Display Response
- Exploits the use of the HTTP protocol beyond HTTP POST and HTTP GET
  - HTTP PUT, HTTP DELETE

REST - not a Standard

- REST is not a standard
  - JSR 311: JAX-RS: The Java™ API for RESTful Web Services
- But it uses several standards:
  - HTTP
  - URL
  - XML/HTML/GIF/JPEG/etc (Resource Representations)
  - text/xml, text/html, image/gif, image/jpeg, etc (Resource Types, MIME Types)

Main Concepts

Nouns (Resources)
unconstrained
i.e., http://example.com/employees/12345



Verbs
constrained
i.e., GET

Representations
constrained
i.e., XML

Resources

- The key abstraction of information in REST is a resource.

- A resource is a conceptual mapping to a set of entities
  - Any information that can be named can be a resource: a document or image, a temporal service (e.g. "today's weather in Los Angeles"), a collection of other resources, a non-virtual object (e.g. a person), and so on

- Represented with a global identifier (URI in HTTP)

  - http://www.boeing.com/aircraft/747

Verbs

- Represent the actions to be performed on resources

- HTTP GET
- HTTP POST
- HTTP PUT
- HTTP DELETE

HTTP GET

- How clients ask for the information they seek.
- Issuing a GET request transfers the data from the server to the client in some representation
- GET http://localhost/books
  - Retrieve all books
- GET http://localhost/books/ISBN-0011021
  - Retrieve book identified with ISBN-0011021
- GET http://localhost/books/ISBN-0011021/authors
  - Retrieve authors for book identified with ISBN-0011021

HTTP PUT, HTTP POST

- HTTP POST creates a resource
- HTTP PUT updates a resource

- POST http://localhost/books/
  - Content: {title, authors[], …}
  - Creates a new book with given properties

- PUT http://localhost/books/isbn-111
  - Content: {isbn, title, authors[], …}
  - Updates book identified by isbn-111 with submitted properties

HTTP DELETE

- Removes the resource identified by the URI

- DELETE http://localhost/books/ISBN-0011
  - Delete book identified by ISBN-0011

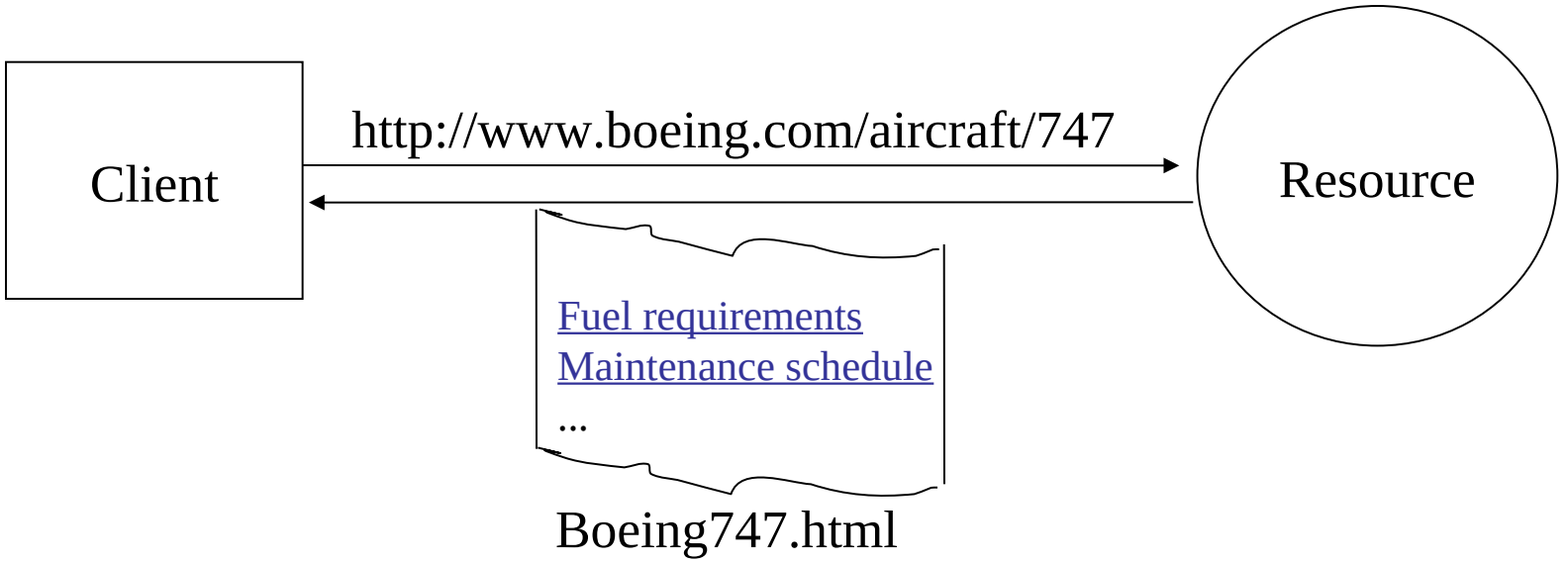
Representations

- How data is represented or returned to the client for presentation.
- Two main formats:
  - JavaScript Object Notation (JSON)
  - XML
- It is common to have multiple representations of the same data

Representations

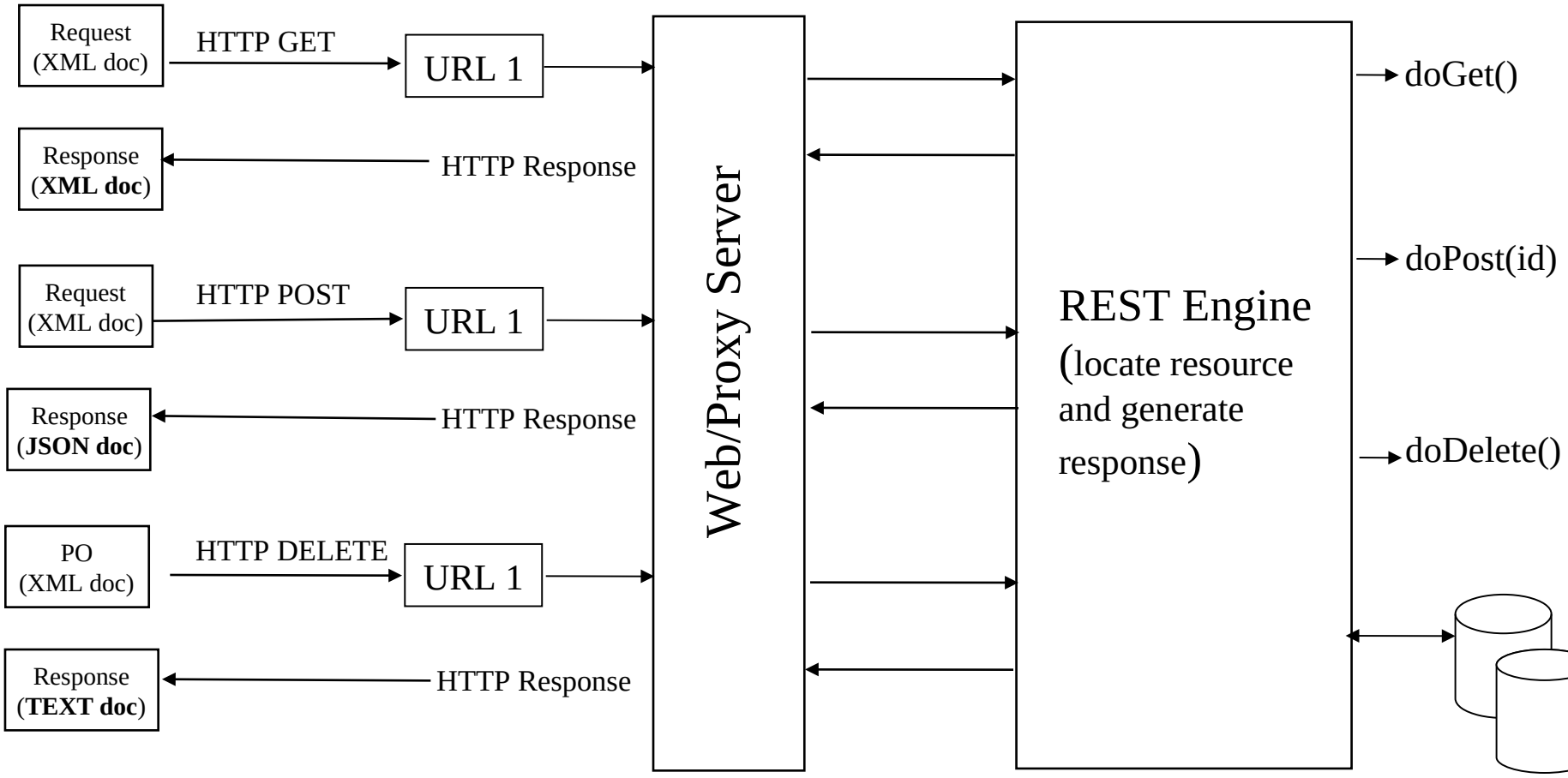
- XML
  - <COURSE>
    - <ID>CS2650</ID>
    - <NAME>Distributed Multimedia Software</NAME>
  - </COURSE>
- JSON
  - {course
    - {id: CS2650}
    - {name: Distributed Multimedia Sofware}
  - }

Why is it called "Representational State Transfer"?



The Client references a Web resource using a URL. A **representation** of the resource is returned (in this case as an HTML document).
The representation (e.g., Boeing747.html) places the client application in a **state**. The result of the client traversing a hyperlink in Boeing747.html is another resource accessed. The new representation places the client application into yet another state. Thus, the client application changes (**transfer**s) state with each resource representation --> Representation State Transfer!

Architecture Style



Real Life Examples

- Google Maps
- Google AJAX Search API
- Yahoo Search API
- Amazon WebServices

REST and the Web

- The Web is an example of a REST system!
- All of those Web services that you have been using all these many years - book ordering services, search services, online dictionary services, etc - are REST-based Web services.
- Alas, you have been using REST, building REST services and you didn't even know it.

REST Implementations

- Restlet
  - http://www.restlet.org/
- Project Zero
  - http://www.projectzero.org
- GlassFish Jersey
  - https://jersey.dev.java.net/
- JBoss RESTeasy
  - http://www.jboss.org/resteasy/

My Project - Web 2.0 IC Card

- Objective
  - Transform the IC Card Application to support REST resources
  - http://www.cs.pitt.edu/icms/iccards
  - http://www.cs.pitt.edu/icms/iccards/001
  - http://www.cs.pitt.edu/icms/groups/1/users

Invocation Model

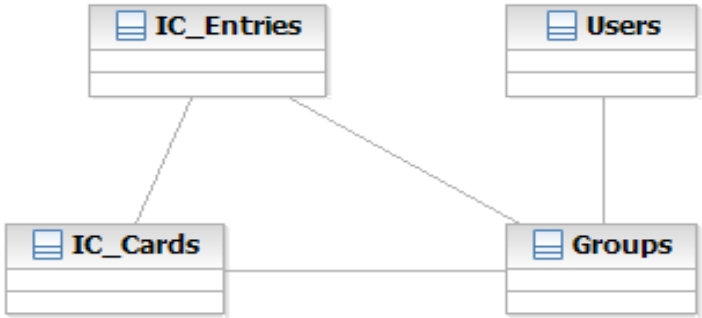
Client

http

ICMS

Channels

http



REST Service

SQL

Database

Tasks

- Design a REST resource model for the IC Card
- Review persistent model (MySql)
- Configure a REST engine (Project Zero)
- Create a new Rich User Interface (DOJO Toolkit)

References

- Representational State Transfer
  http://en.wikipedia.org/wiki/Representational_State_Transfer

- Roy Fieldings Thesis

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