Lesson 1 – BPEL Introduction

Business process engineering

Module 1 - Web services

Unit 1 – Web protocols

Ernesto Damiani

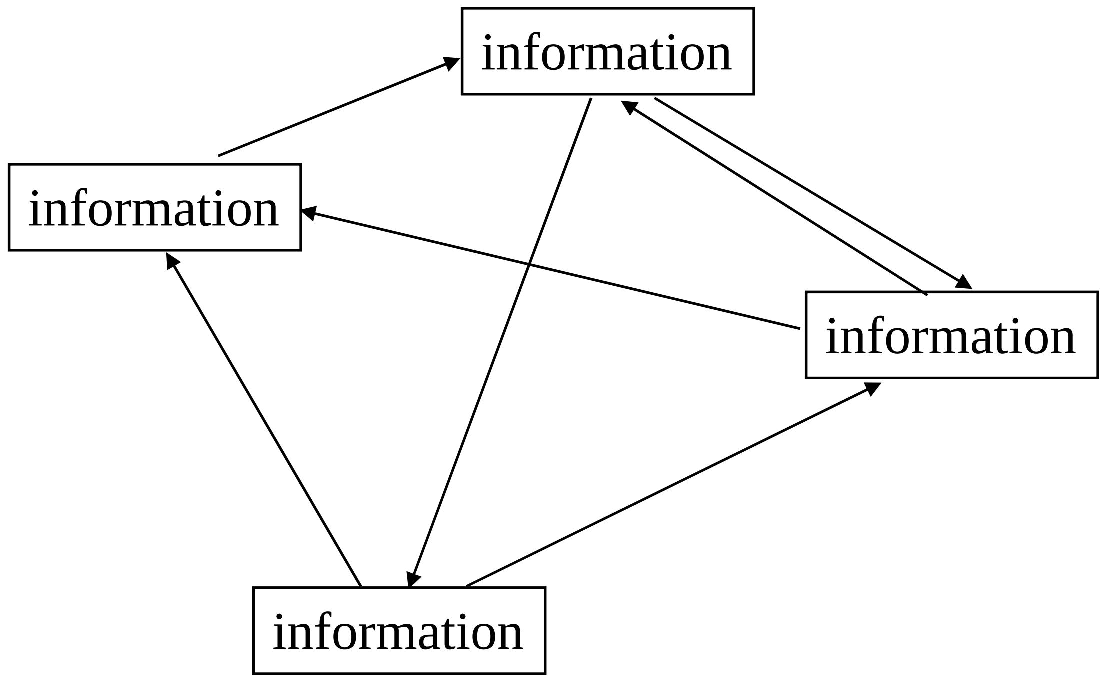
Università di Milano

Web services

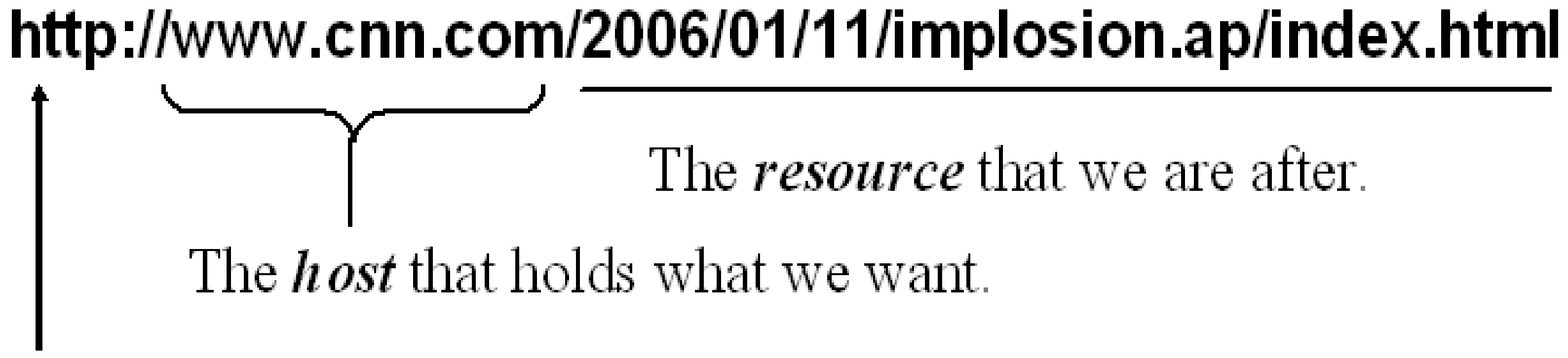
- A Web Service is a software implementation of a resource, identified by a URL, reached using internet protocols

The Web

- The Web is a network of information that can be traversed in many ways



What’s in a Name?



http://www.cnn.com/2006/01/11/implosion.ap/index.html

The ***resource*** that we are after.

The ***host*** that holds what we want.

The ***scheme*** identifies the protocol that is to be used; in this example, the HyperText Transfer Protocol.

There are 51 other schemes, among them

- https (HyperText Transfer Protocol Secure)
- ftp (File Transfer Protocol)
- urn (Uniform Resource Names)

HTTP (version 1.1) is by far the most commonly used scheme.

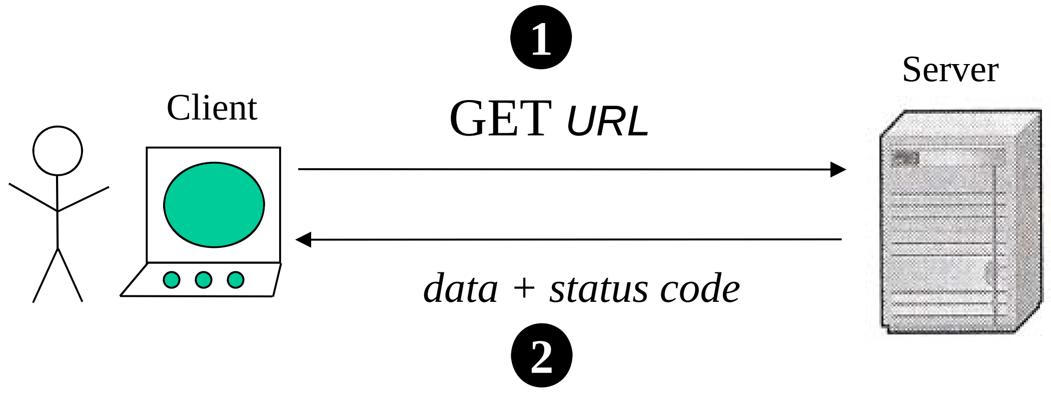
For all practical purposes, HTTP 1.1 is the "Web Protocol."

The Web API

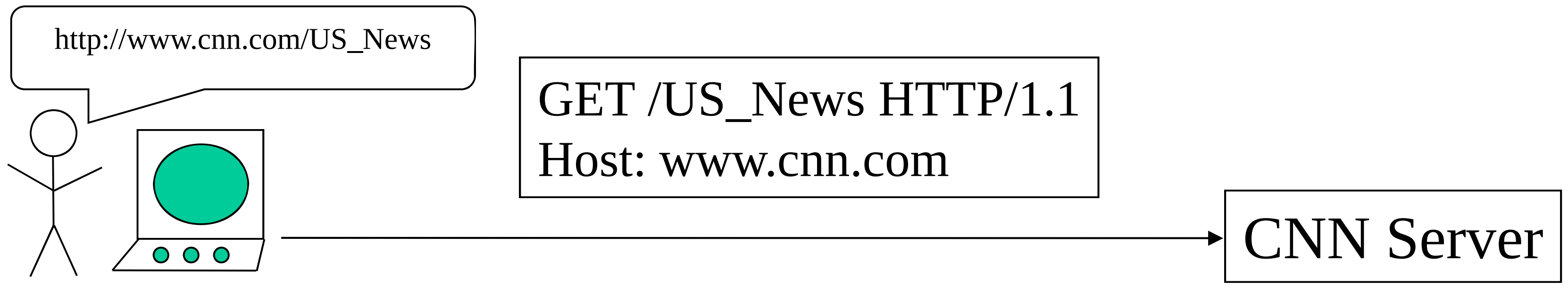
- HTTP provides a simple set of operations. Amazingly, all Web exchanges are done using this simple HTTP API
  - **GET** = "give me some stuff" (Retrieve)
  - **POST** = "here's some better stuff" (Update)
  - **PUT** = "here's some new stuff" (Create)
  - **DELETE** = "delete that stuff" (Delete)
- A few simple rules allow you to create tremendous complexity

Retrieving Information

- The server responds with, not only the data (the Web page), but also a result code (200 means everything is OK)



HTTP GET (1)



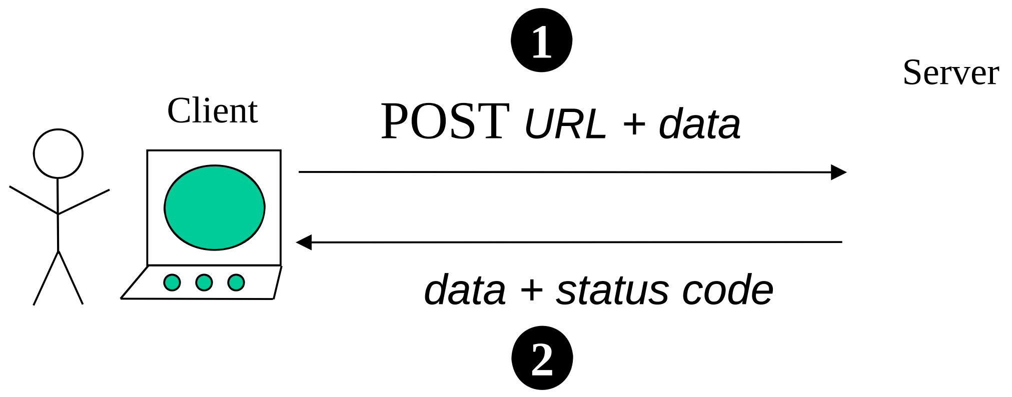
GET /US_News HTTP/1.1
Host: www.cnn.com

HTTP GET (2)

- The user types in at his browser:
http://www.cnn.com/US_News
- The browser software creates an HTTP header
  - The HTTP header identifies:
    - The desired action: GET ("get me some stuff")
    - The target machine (www.cnn.com)
    - The resource (US_News)
    - The version of HTTP being used (version 1.1)

Updating Information

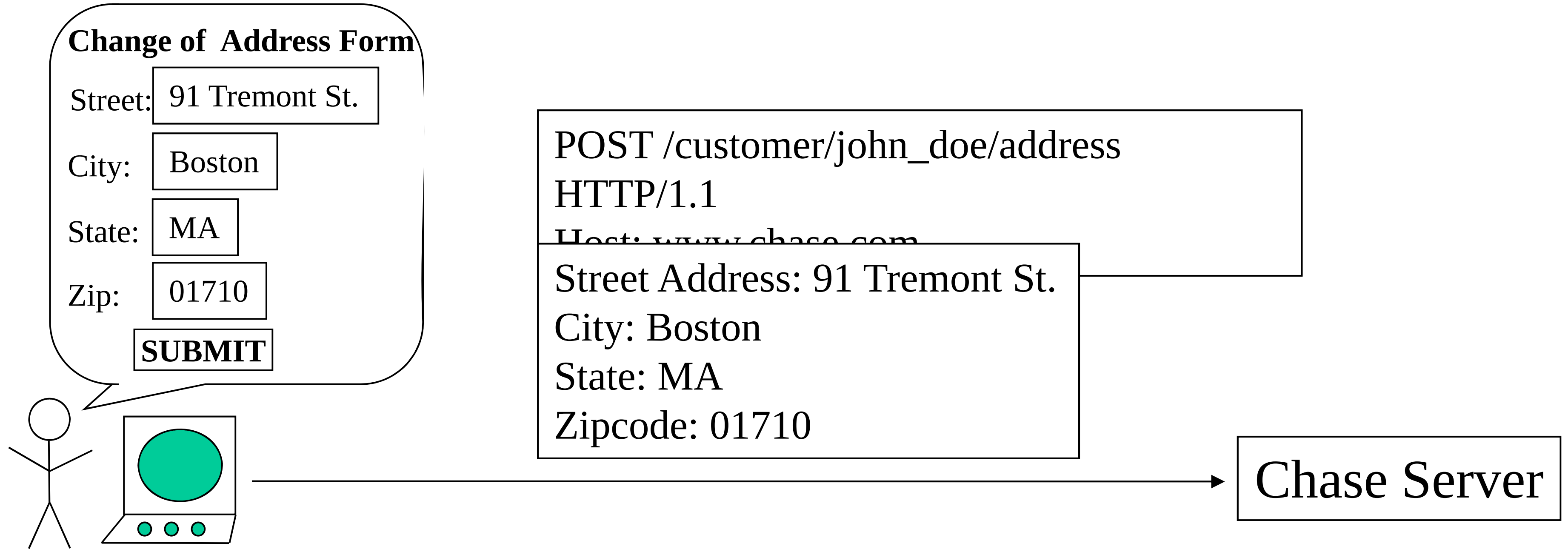
- The client provides information. The server responds with a result code (200 means everything is OK)



HTTP POST (1)

- The user fills in the Web page's form (Chase Bank's Change of Address Form)
- The browser software creates an HTTP header, and a payload which is comprised of the form data
  - The HTTP header identifies:
    - The desired action: POST ("here's my new address")
    - The target machine (www.chase.com)
    - The resource (customer/john_doe/address)
  - The payload contains:
    - The data being POSTed (the form data)

HTTP POST (2)



Change of Address Form

Street:	91 Tremont St.
City:	Boston
State:	MA
Zip:	01710

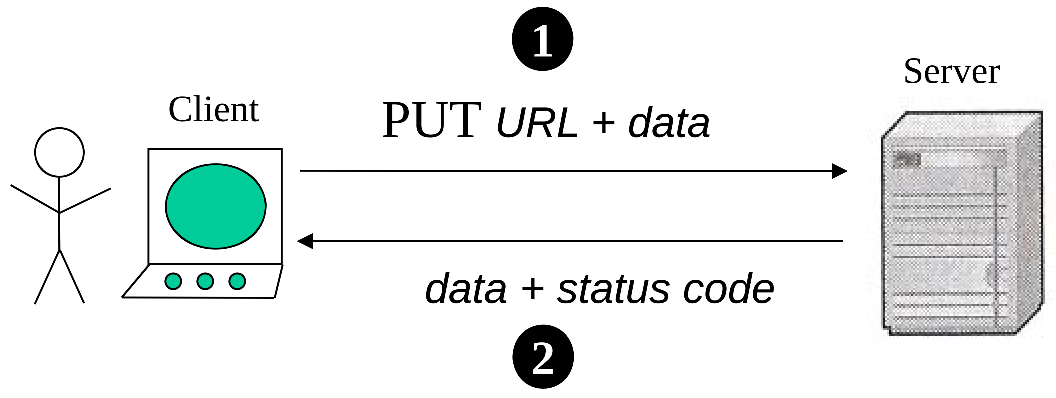
SUBMIT

POST /customer/john_doe/address
HTTP/1.1
Host: www.chase.com

Street Address: 91 Tremont St.
City: Boston
State: MA
Zipcode: 01710

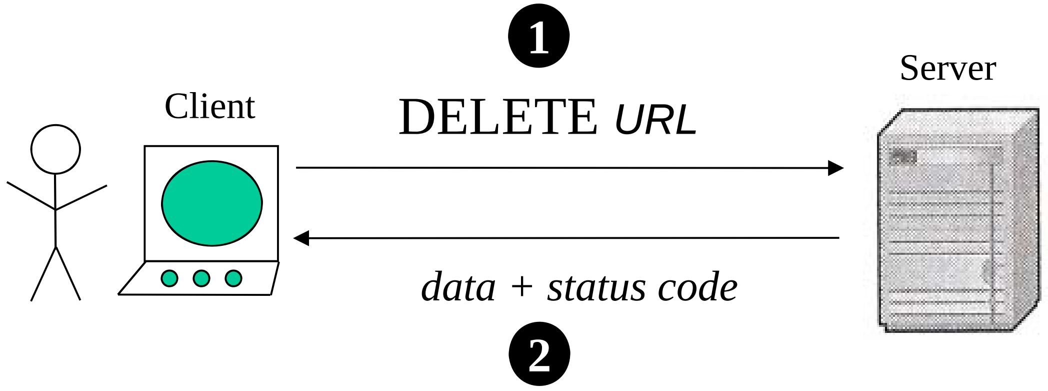
Providing Information

- **POST** is used to update existing information on the server.
- **PUT** is used to make new information available on the server



Removing Information

- The client requests that the server remove information identified by the URL. Typically a server will return an acknowledgement that it has deleted the requested data

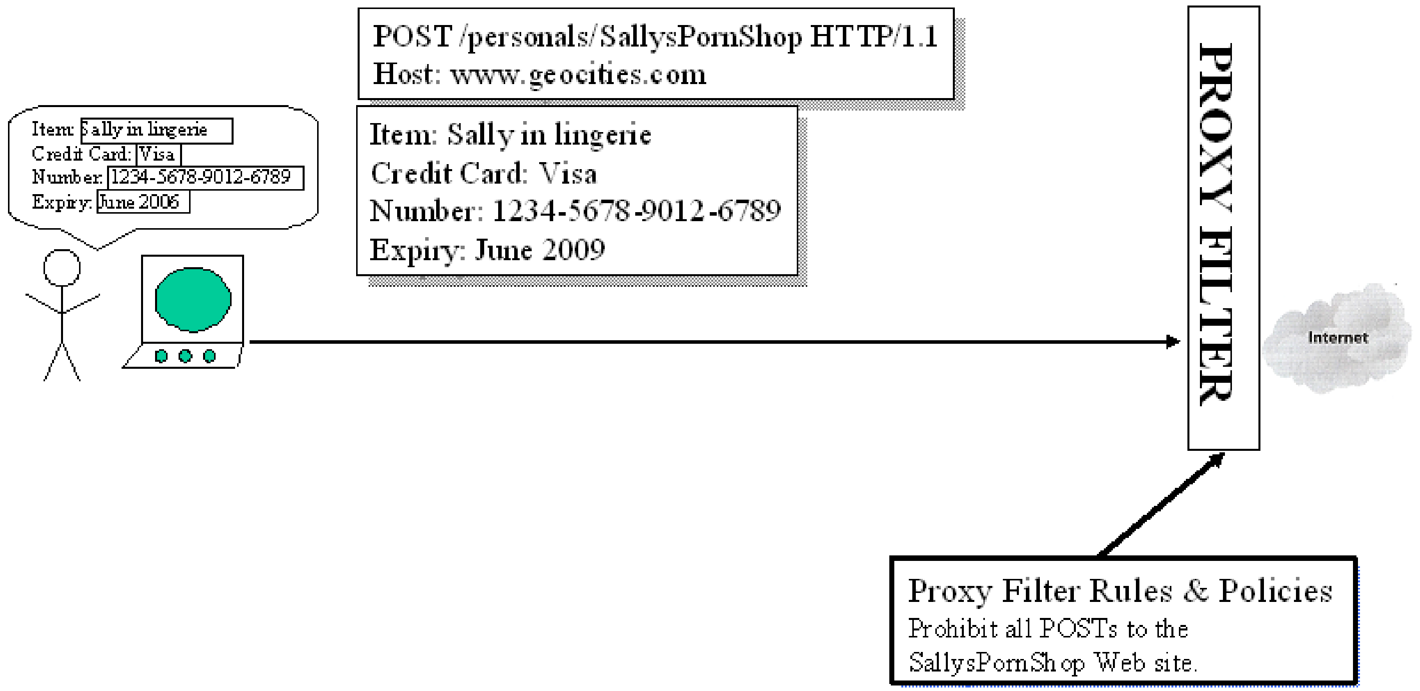


Basic Web Components

- **Firewalls** and **Proxy Filters**: these components decide what HTTP messages get out, and what get in
  - These components enforce Web **security**
- **Routers**: these components decide where to send HTTP messages
  - These components manage Web **scaling**
- **Caches**: these components decide if a saved copy can be used
  - These components increase Web **speed**

Firewalls and Proxy Filters

- The proxy filter decides whether an HTTP message should pass
  - This message is rejected!

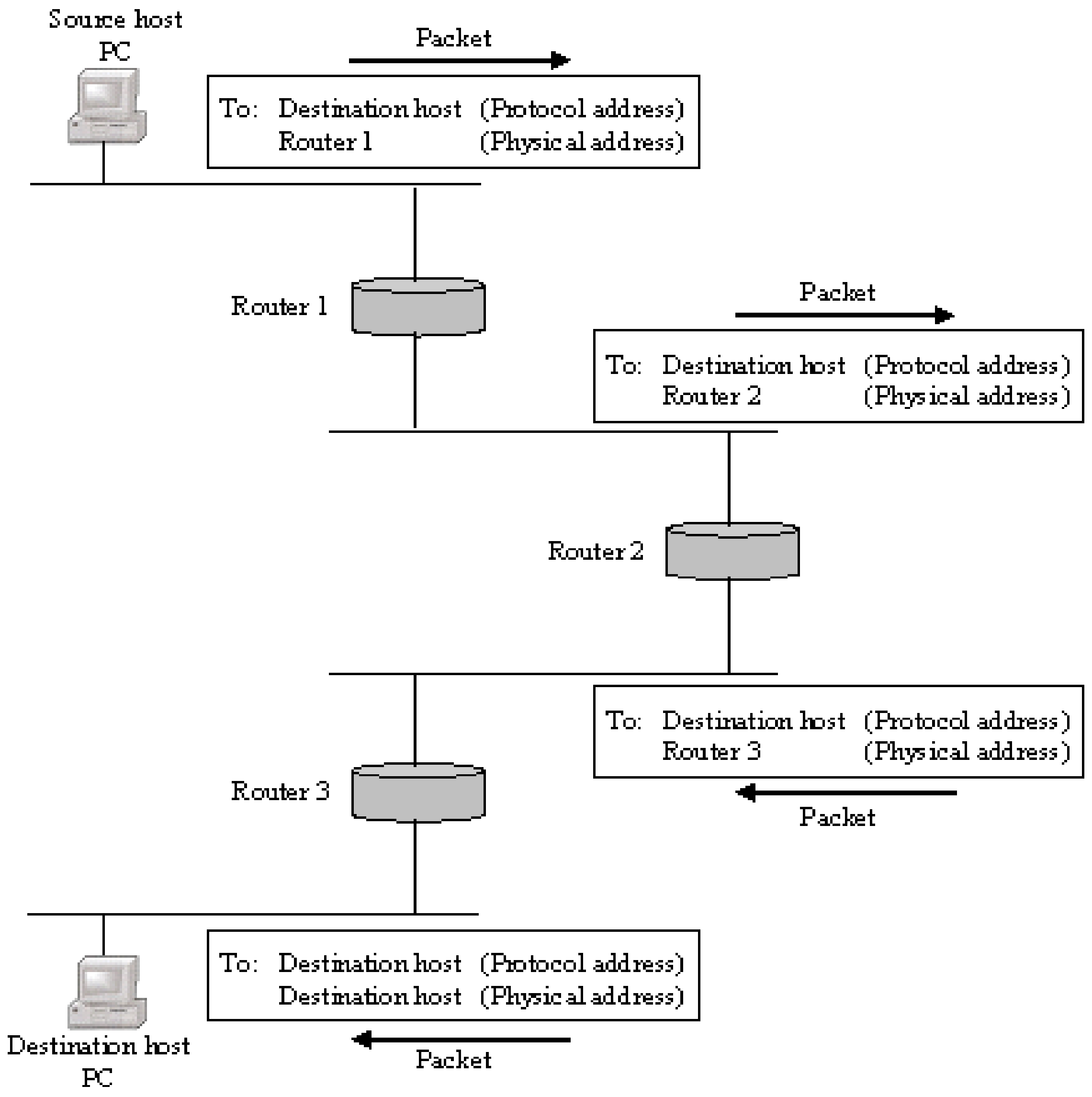


```
POST /personals/SallysPornShop HTTP/1.1
Host: www.geocities.com
```

Item: Sally in lingerie
Credit Card: Visa
Number: 1234-5678-9012-6789
Expiry: June 2009

Proxy Filter Rules & Policies
Prohibit all POSTs to the SallysPornShop Web site.

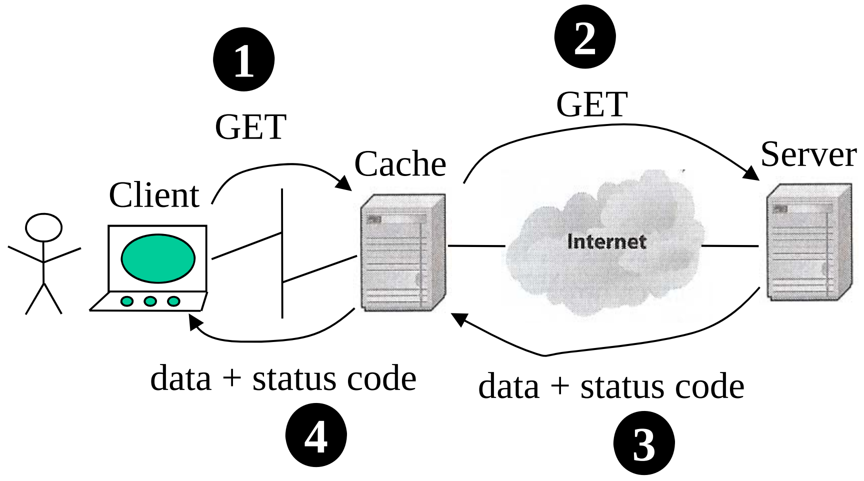
Routers



Source host
PC
Packet
To: Destination host (Protocol address)
Router 1 (Physical address)
Router 1
Packet
To: Destination host (Protocol address)
Router 2 (Physical address)
Router 2
To: Destination host (Protocol address)
Router 3 (Physical address)
Packet
Router 3
To: Destination host (Protocol address)
Destination host (Physical address)
Packet
Destination host
PC

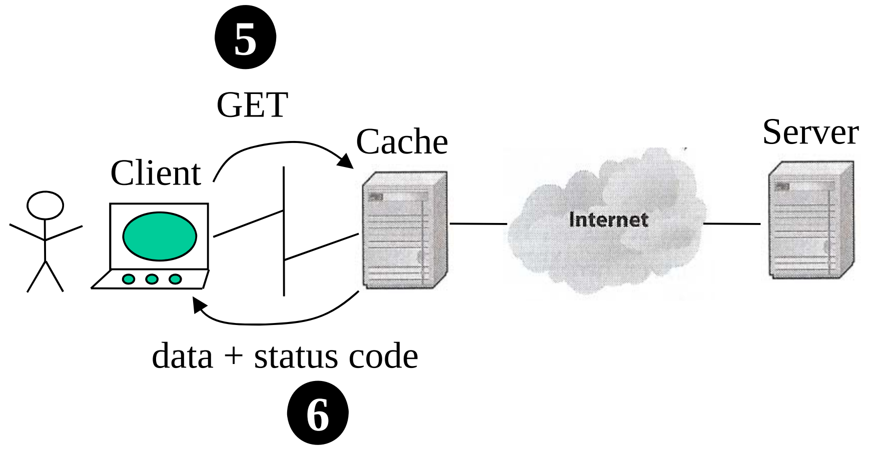
Cache (1)

- First access, goes all the way back to the server with the data (origin server)



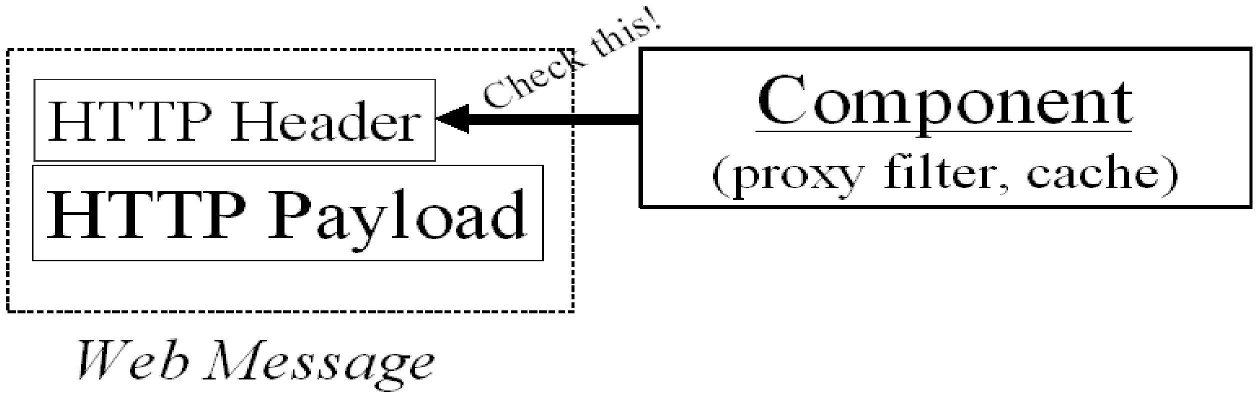
Cache (2)

- Next access, the cache returns the data

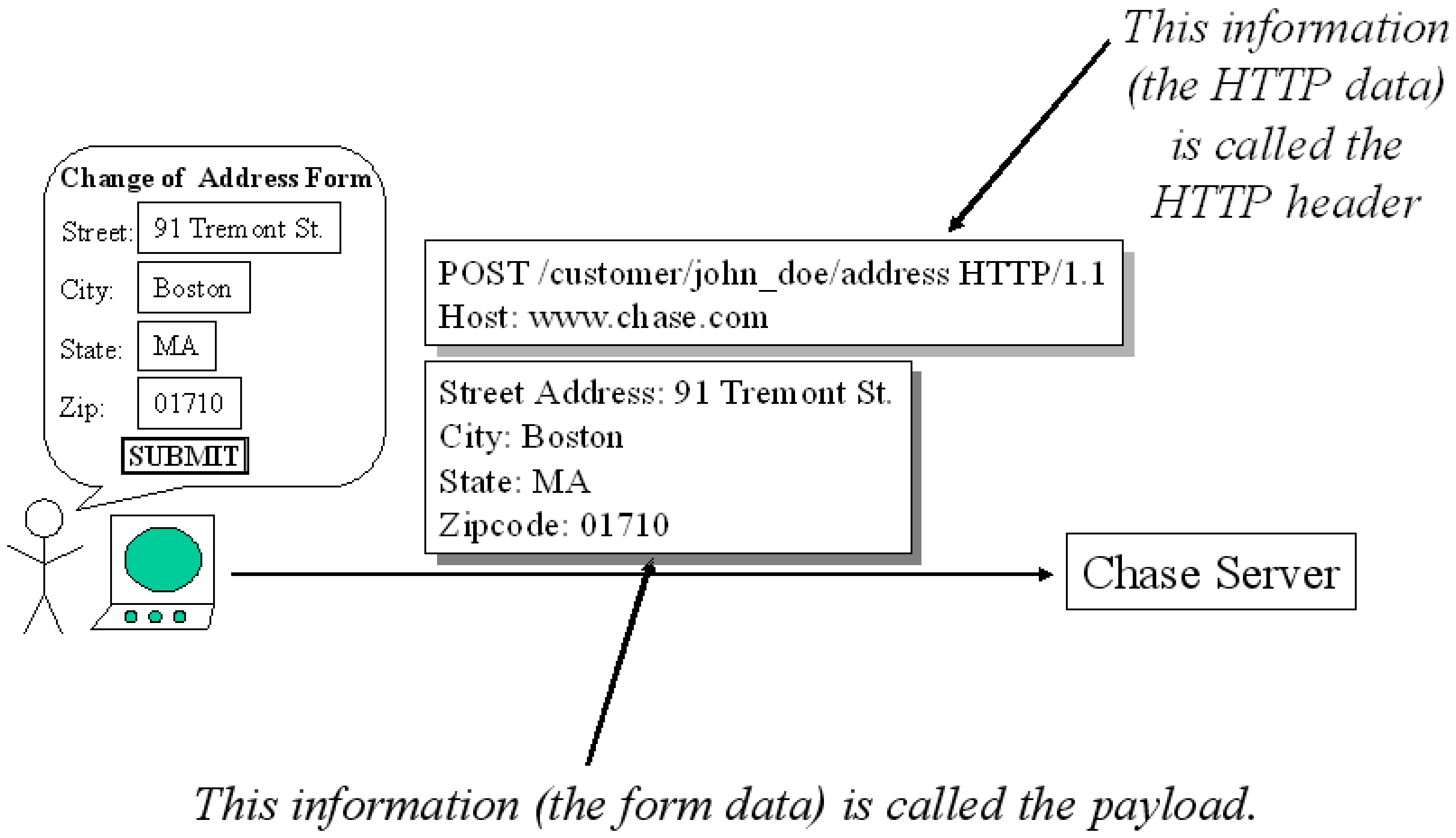


Proxy Filters and Caches

- Proxy filters and caches operate using only information found in the HTTP header!
  - (Firewalls and routers use lower-level information, such as IP addresses)



Header and Payload


Change of Address Form
Street: 91 Tremont St.
City: Boston
State: MA
Zip: 01710
SUBMIT
Chase Server
This information (the HTTP data) is called the HTTP header
This information (the form data) is called the payload.

```
POST /customer/john_doe/address HTTP/1.1
Host: www.chase.com
```

```
Street Address: 91 Tremont St.
City: Boston
State: MA
Zipcode: 01710
```

Reasons for Basing Decisions Solely on the HTTP Header

- The content of the HTTP header is well-defined (standard semantics)
- Conversely, the HTTP payloads change from request to request
- Web components cannot make sense of all the different kinds of information that may occur in payloads
- Web components **never** peek inside the message payload