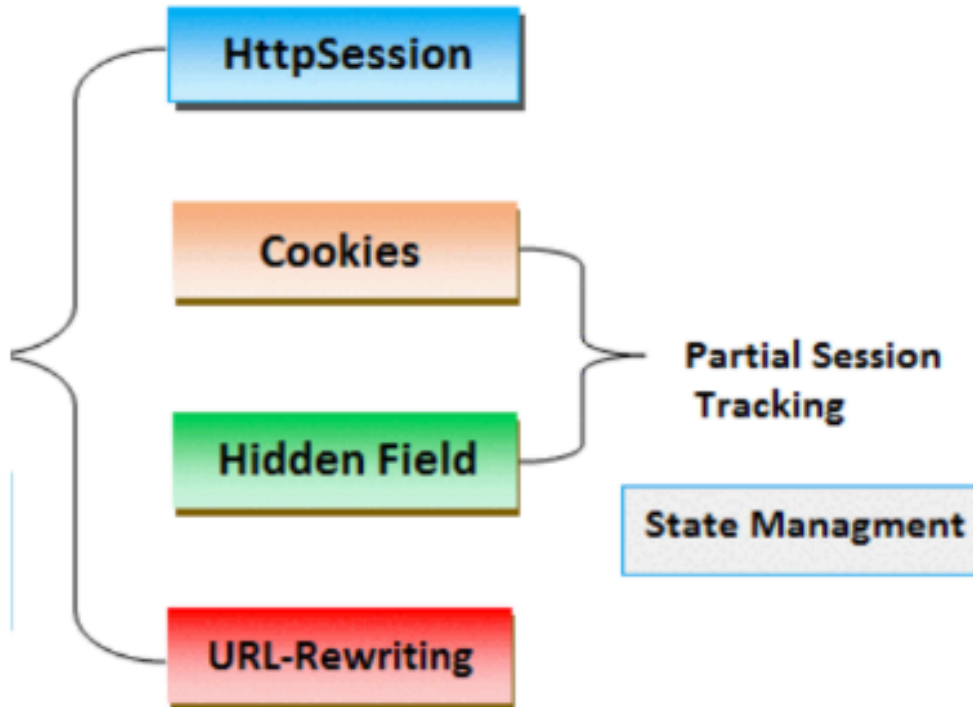



Sessions

Sessions Introduction



The diagram illustrates various session tracking techniques. On the left, a large curly bracket groups four colored boxes: 'HttpSession' (blue), 'Cookies' (orange), 'Hidden Field' (green), and 'URL-Rewriting' (red). On the right, a smaller curly bracket groups 'Cookies' and 'Hidden Field', with a line pointing to the text 'Partial Session Tracking'. Below this, a box labeled 'State Management' is connected to the 'HttpSession' box.

Common techniques for implementing sessions.



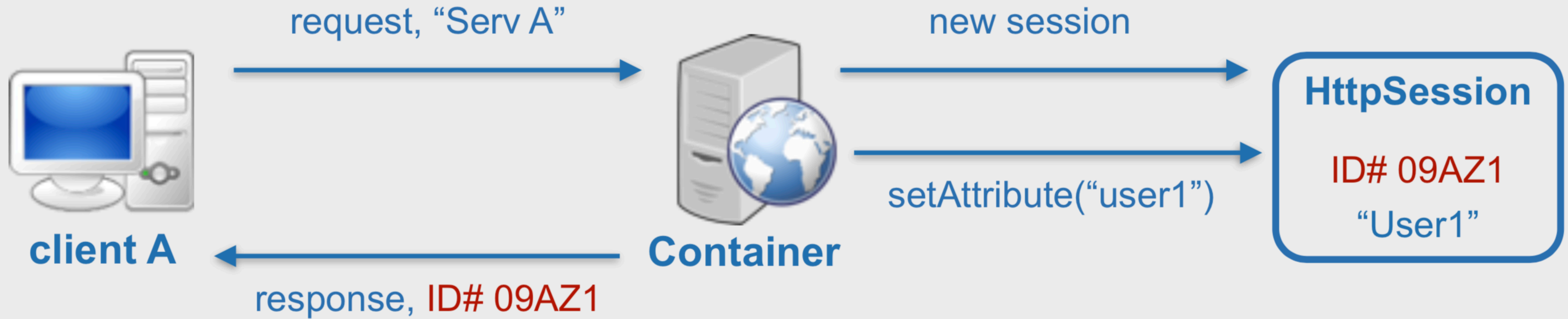
How to Make an Application out of a Web Page?

- On the internet, a web page is a web page is a web page...
 - If you surf from ./page1.html to ./page2.html these are two unique requests.
 - The server doesn't know anything about the fact that both pages are visited by the same user.
- Sessions are the technique used to logically group several requests into a "group" (called a session)
 - If you start a session, the server will know that it's still the same user who surfed from ./page1.html to ./page2.html

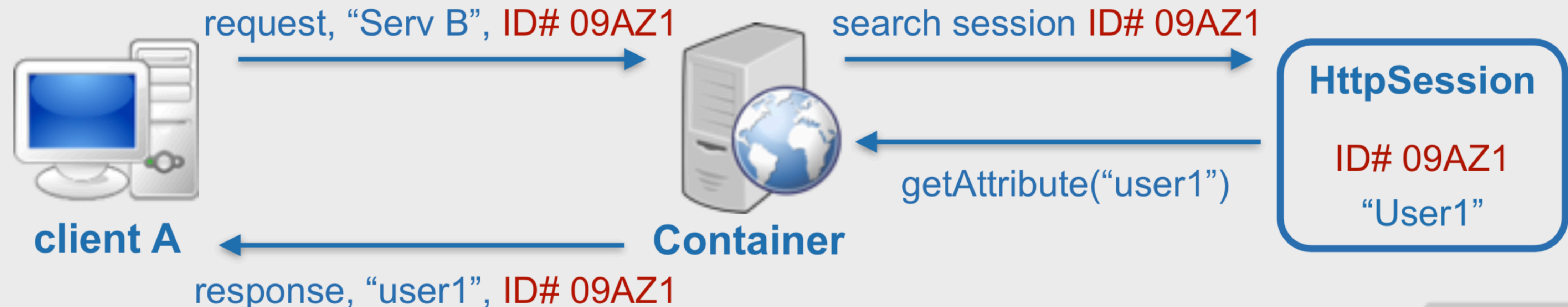


Session Tracking

First Request

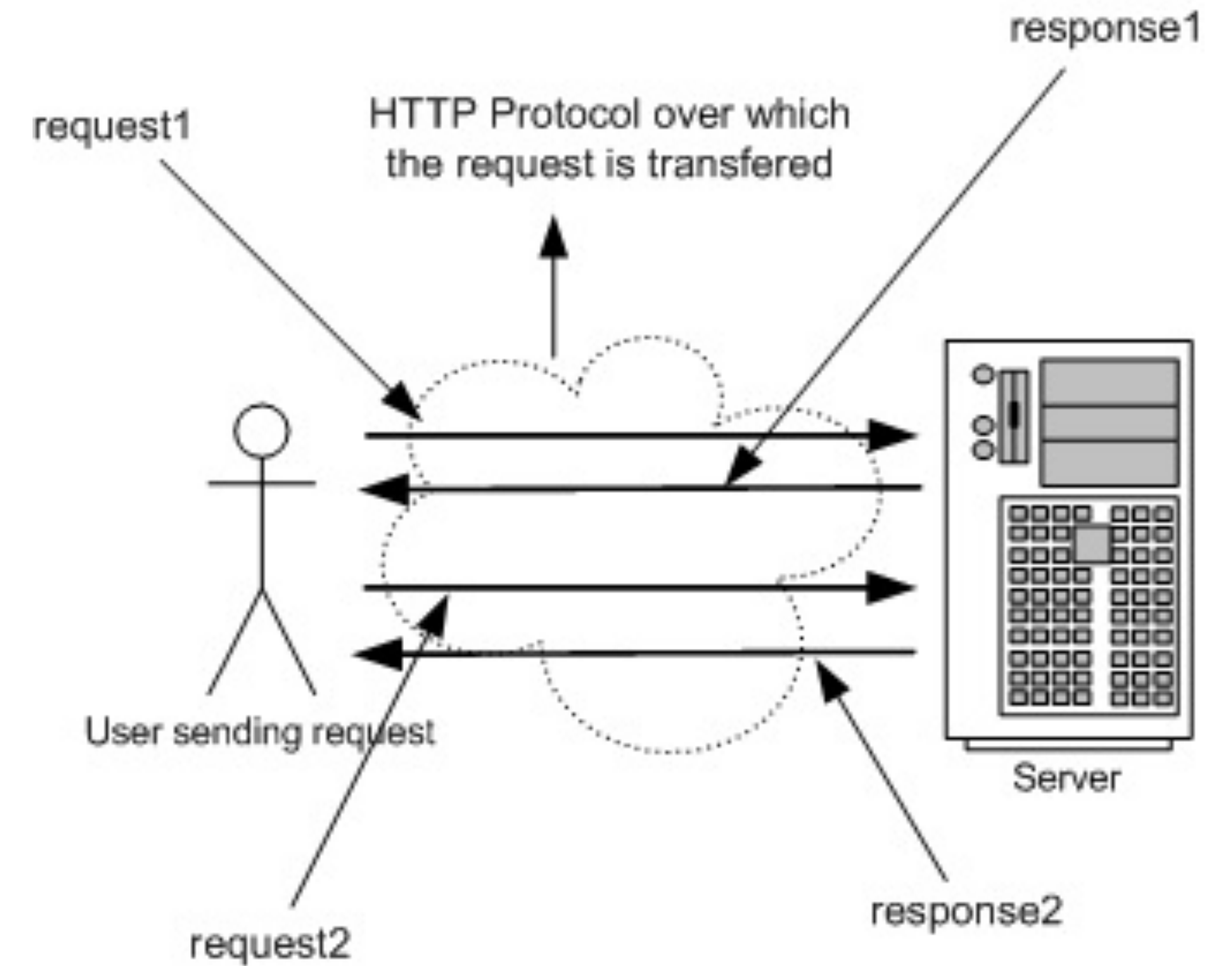


Subsequent Request



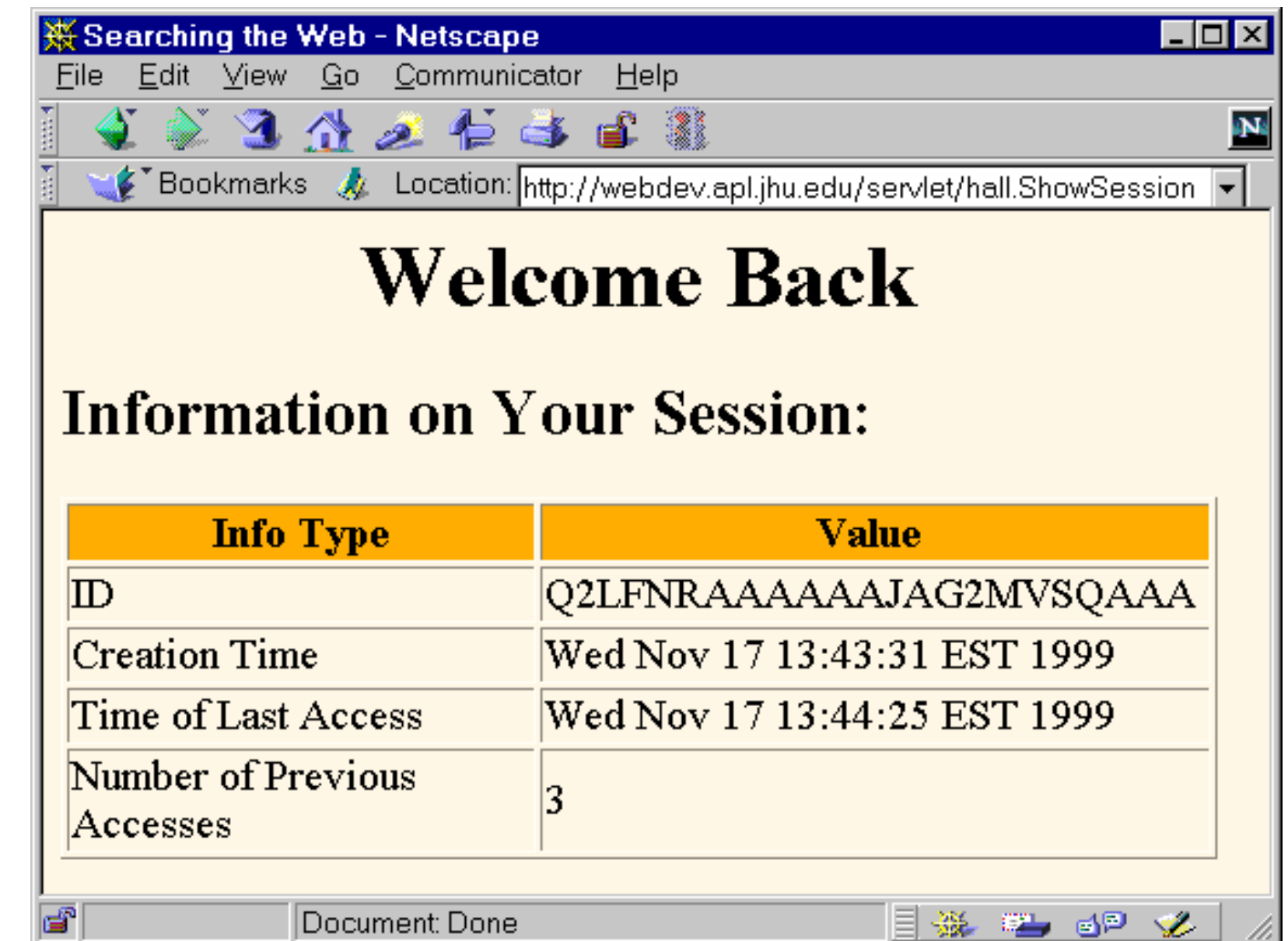
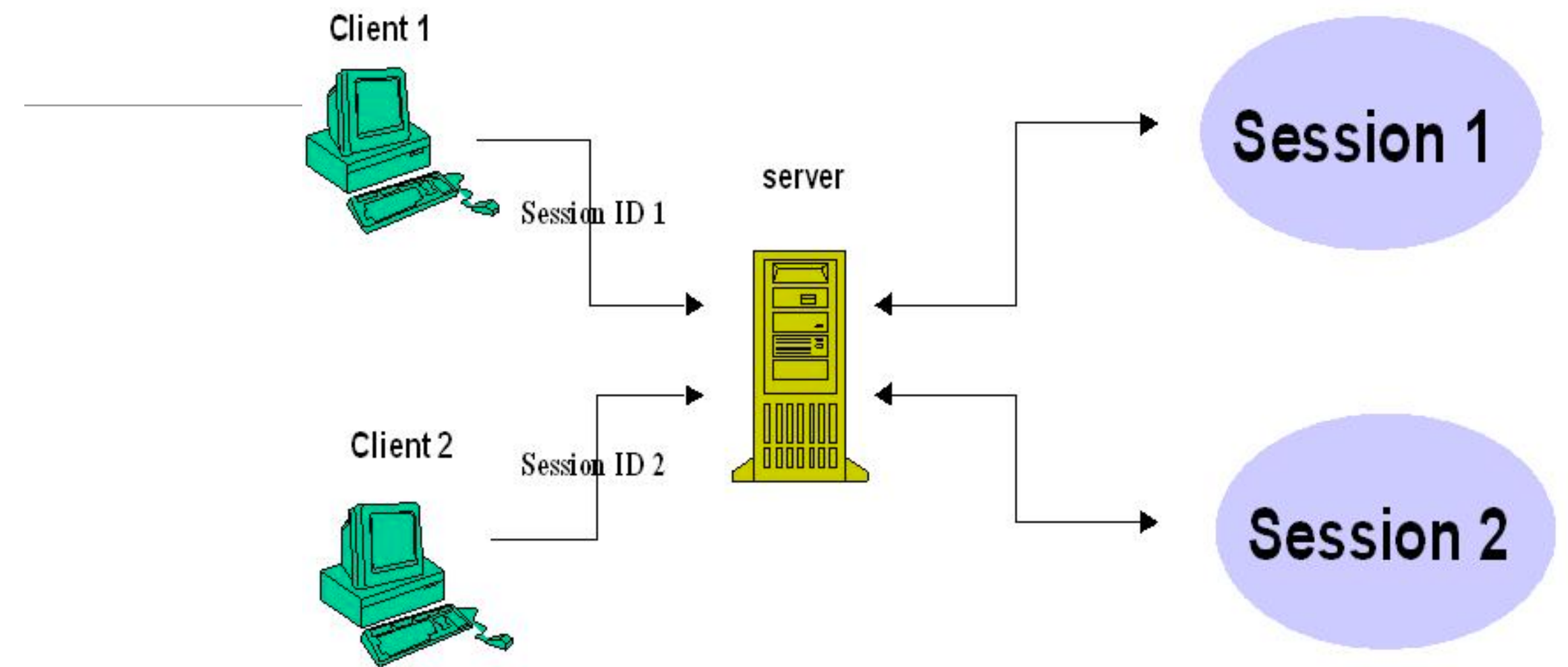
Sessions

- HTTP itself is “stateless”
 - no state stored on the server between requests from the same client
- but many web apps are stateful
 - necessary to connect requests from the same user / browser / browser-window, e.g. shopping cart, appointments calendar etc...
- *Session*
 - multiple requests performed in a stateful context
- *Session tracking*
 - technique that allows sessions in stateless environments



- User surfs to http://demo.com
- Server (on 1st request / if no sessionID stored on client)
 - generates unique session id, which is mapped to ...
 - ... a session-object
 - stored in memory (lost on shutdown), in a file or in database
 - can contain anything (list of articles, game state, counters, ...)
 - Session id is added to the response
- from now on:
 - each subsequent request from the same user (browser) must contain the session id ...
 - ... which is used by the server to map to the session-object
- No data gets stored on the client, except SessionID

Session Tracking



Session Tracking Techniques

- Cookie
- Hidden Form Field
- URL Rewriting
- Json Web Token (JWT)

Cookies

First Response



client A



Http Response

```
HTTP/1.1 200 OK
Location: http://www.abcd.com/login
Set-Cookie: JSESSIONID=09AZ1
Domain=.abcd.com;path=/;HttpOnly
.....
```



Container



Subsequent Requests



client A



Http Request

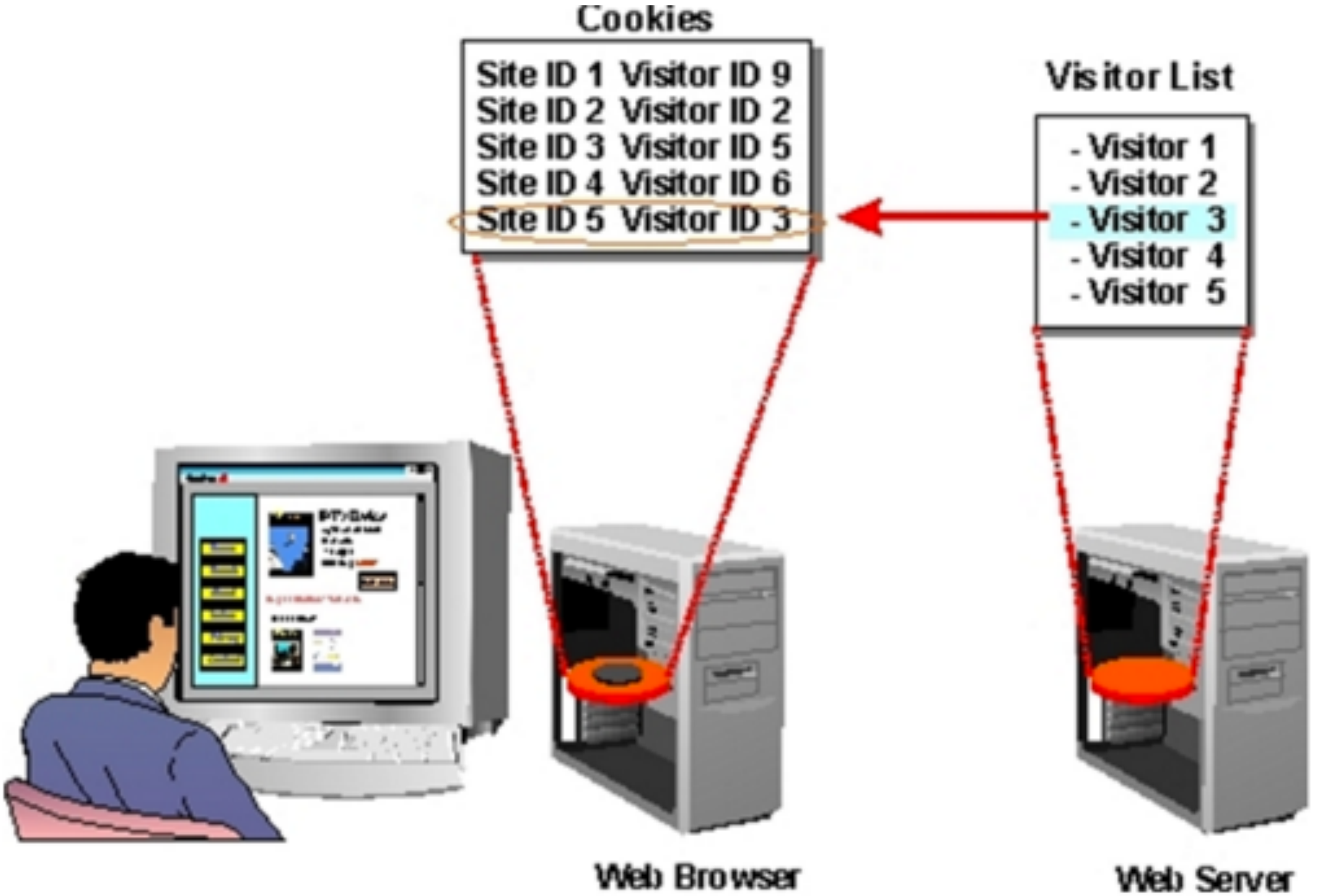
```
POST/login.do HTTP/1.1
Host: www.abcd.com
Cookie: JSESSIONID=09AZ1
.....
```



Container

Cookies

1. Server creates a cookie with session-id on first request
2. Server maps id to a new user-specific session object
3. The session-id is sent to the client with the first response
4. ..and automatically added by the browser on each further request (to the same address/domain/...)
5. Server receives request + cookie with session-id
6. Server maps session-id to session-object



cookie (in browser)

The screenshot shows the browser's developer tools with the 'Storage' tab selected. The 'Cookies' section is expanded, showing a table of cookies for the URL 'http://localhost:3000'. The selected cookie is 'donation-web' with domain 'localhost' and path '/'. The right-hand pane shows the details of this cookie, including its value, creation time, domain, expiration, and attributes like 'HostOnly', 'HttpOnly', 'Secure', and 'sameSite'.

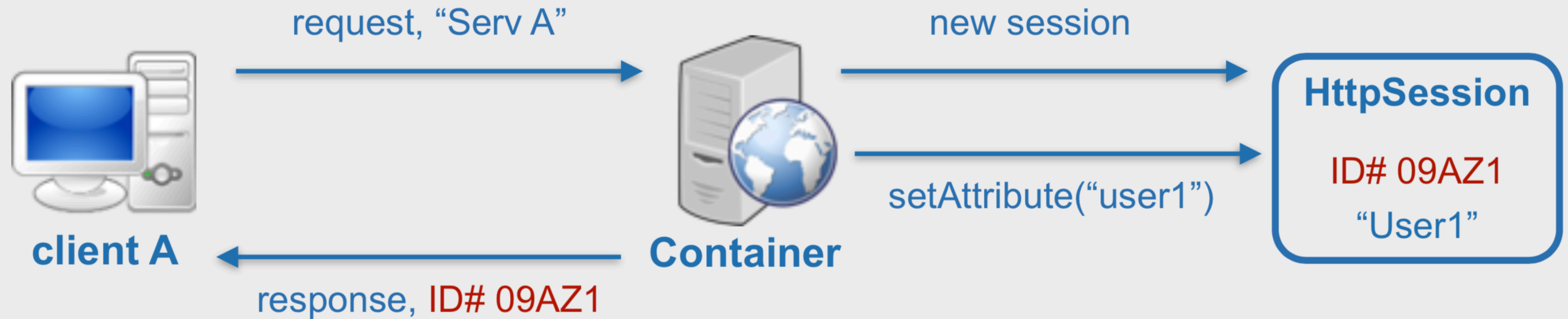
Name	Domain	Path	Expires on	Last accessed on	Value
donation-web	localhost	/	Thu, 31 Jan 2019 07:01:14 GMT	Wed, 30 Jan 2019 07:01:14 GMT	Fe26.2**78d4c93c186076e4cd...LNHdXmQBk2v4cTgKTZKEKvBWU

Cookie Details:

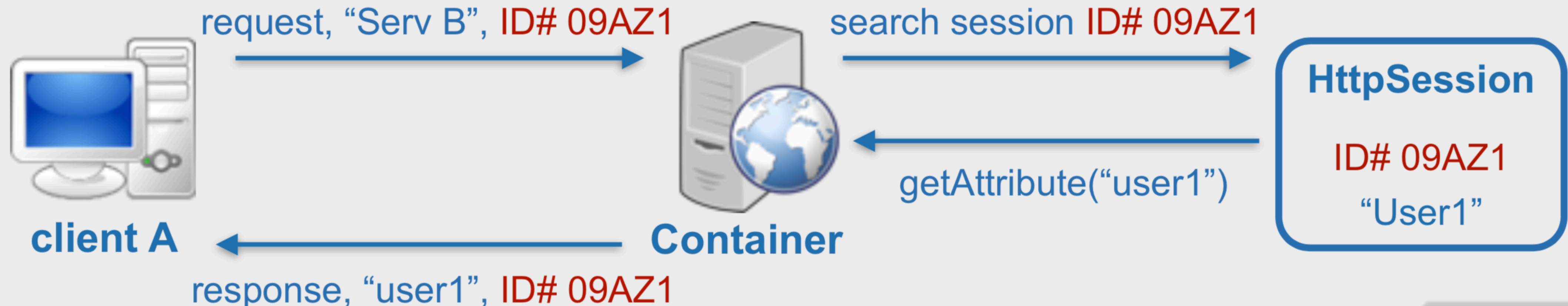
- Data:**
 - donation-web: "Fe26.2**78d4c93c186076e4cd...LNHdXmQBk2v4cTgKTZKEKvBWU"
 - CreationTime: "Wed, 30 Jan 2019 07:01:14 GMT"
 - Domain: "localhost"
 - Expires: "Thu, 31 Jan 2019 07:01:14 GMT"
 - HostOnly: true
 - HttpOnly: true
 - LastAccessed: "Wed, 30 Jan 2019 07:01:14 GMT"
 - Path: "/"
 - Secure: false
 - sameSite: "Strict"
- Parsed Value:**
 - donation-web: Array
 - 0: "Fe26.2"
 - 1: ""
 - 2: "78d4c93c186076e4cd271c2b1d33e5a1548cf79134789aaee69cd0a4389b7174"
 - 3: "EV_4-Ds7qYD3Qqz3YxyPIQ"
 - 4: "TtPdRqA3xzll5DlVBJjXG3nr9gMPiXx_2ThreofArK-7HMKL18neAgUCrmibC99K"
 - 5: ""
 - 6: "28a6a80b7dece305e6cfa1c7661d50cbfe1220f121e1c5687c7a72289c405ebc"
 - 7: "zTN-pcJx0oYtzS4oQhLNHdXmQBk2v4cTgKTZKEKvBWU"
 - length: 8
 - __proto__: Array

URL Rewrite

First Request



Subsequent Request



URL Rewrite

- Server adds the session-id to all links the user can follow
 - `http://server/myhome`
- is changed to
 - `http://server/myhome?sessionid=123`
- session-id must be dynamically added
 - functionality usually offered by scripting frameworks

Hidden Form Fields

- In HTML, we can define "hidden" fields in a form
 - `<input type="hidden" name="sessionid" value="123">`
- These fields are not visible and cannot be changed by the client
- Usage:
 - server creates a session-object for each client and generates a unique ID
 - When HTML documents are created and sent back, the hidden form field is automatically generated containing the actual ID
 - Upon form submit, the session ID is automatically sent back to the server
 - The server can associate this call with an already existing session

Hidden Form Filed Example


Donation Donate Report Settings Logout

Select Amount ▾

Paypal

Direct

Donate



A screenshot of a web application's donation page. At the top is a dark navigation bar with links for 'Donation', 'Donate', 'Report', 'Settings', and 'Logout'. The main content area features a form on the left with a 'Select Amount' dropdown menu, two radio buttons for 'Paypal' and 'Direct', and a blue 'Donate' button. To the right of the form is a cartoon image of Homer Simpson with his arms raised in a celebratory gesture. Below the form is a progress bar with a small teal segment on the left.

```
<form action="/donate" method="POST">
  <input type="hidden" name="userID" value="2354515">
  <div class="ui dropdown" name="amount">
    <input type="hidden" name="amount">
    <div class="text">Select Amount</div>
    <i class="ui dropdown icon"></i>
    <div class="menu">
      <div class="item">50</div>
      <div class="item">100</div>
      <div class="item">1000</div>
    </div>
  </div>
  <div class="grouped inline fields">
    <div class="field">
      <div class="ui radio checkbox">
        <input type="radio" name="method" value="paypal">
        <label>Paypal</label>
      </div>
    </div>
    <div class="field">
      <div class="ui radio checkbox">
        <input type="radio" name="method" value="direct">
        <label>Direct</label>
      </div>
    </div>
  </div>
  <button class="ui blue submit button">Donate</button>
</form>
```

Hidden Form Filed Example



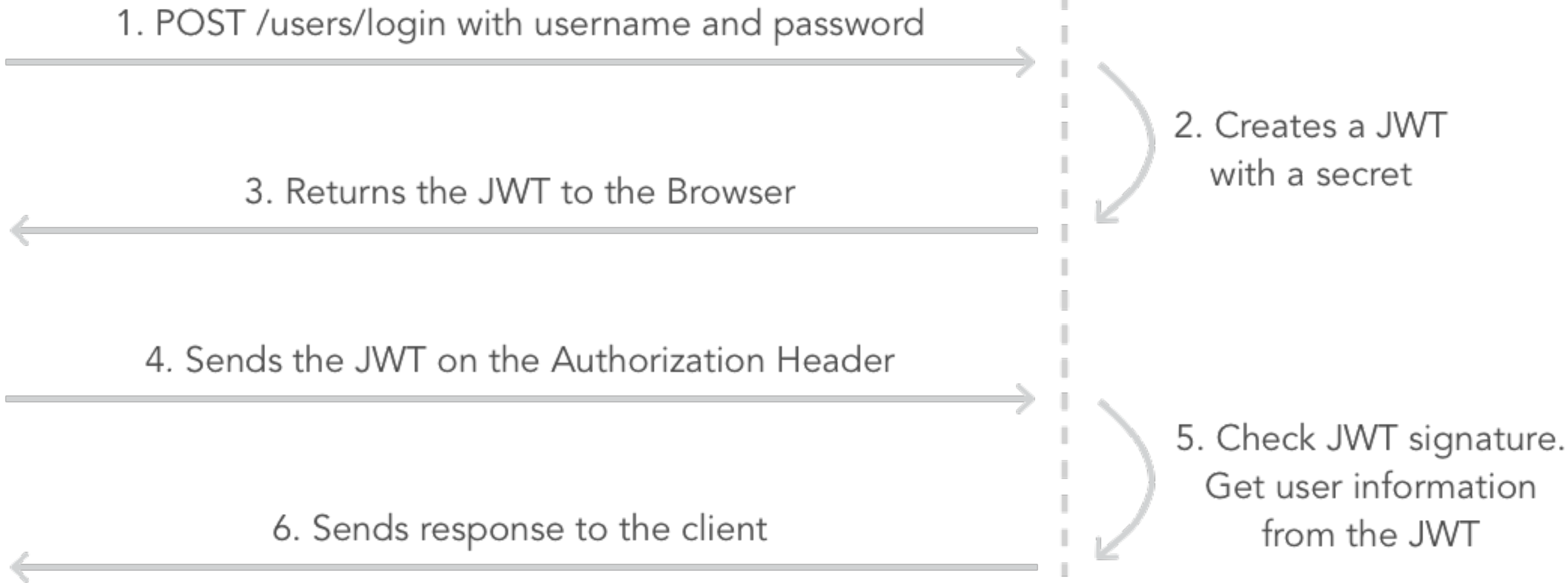
```
<form action="/donate" method="POST">
  <input type="hidden" name="userID" value="2354515">
  <div class="ui dropdown" name="amount">
    <input type="hidden" name="amount">
    <div class="text">Select Amount</div>
    <i class="ui dropdown icon"></i>
    <div class="menu">
      <div class="item">50</div>
      <div class="item">100</div>
      <div class="item">1000</div>
    </div>
  </div>
  <div class="grouped inline fields">
    <div class="field">
      <div class="ui radio checkbox">
        <input type="radio" name="method" value="paypal">
        <label>Paypal</label>
      </div>
    </div>
  </div>
</form>
```

Json Web Token

- An open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.
 - **Compact:** Because of its smaller size, JWTs can be sent through an URL, POST parameter, or inside an HTTP header.
 - **Self-contained:** The payload contains all the required information about the user, avoiding the need to query the database more than once.
- **Authentication:** Once the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token.
- **Information Exchange:** JSON Web Tokens are a good way of securely transmitting information between parties, because they can be signed.

Browser

Server



Web Frameworks

- Cookies generally preferred.
- However, framework may try to ‘abstract away’ specific session management technology, and deliver simpler abstraction to the programmer
- Framework may in fact be able to switch between different techniques depending on circumstances.