



ASP.NET AJAX

Lương Trần Hy Hiến

Agenda

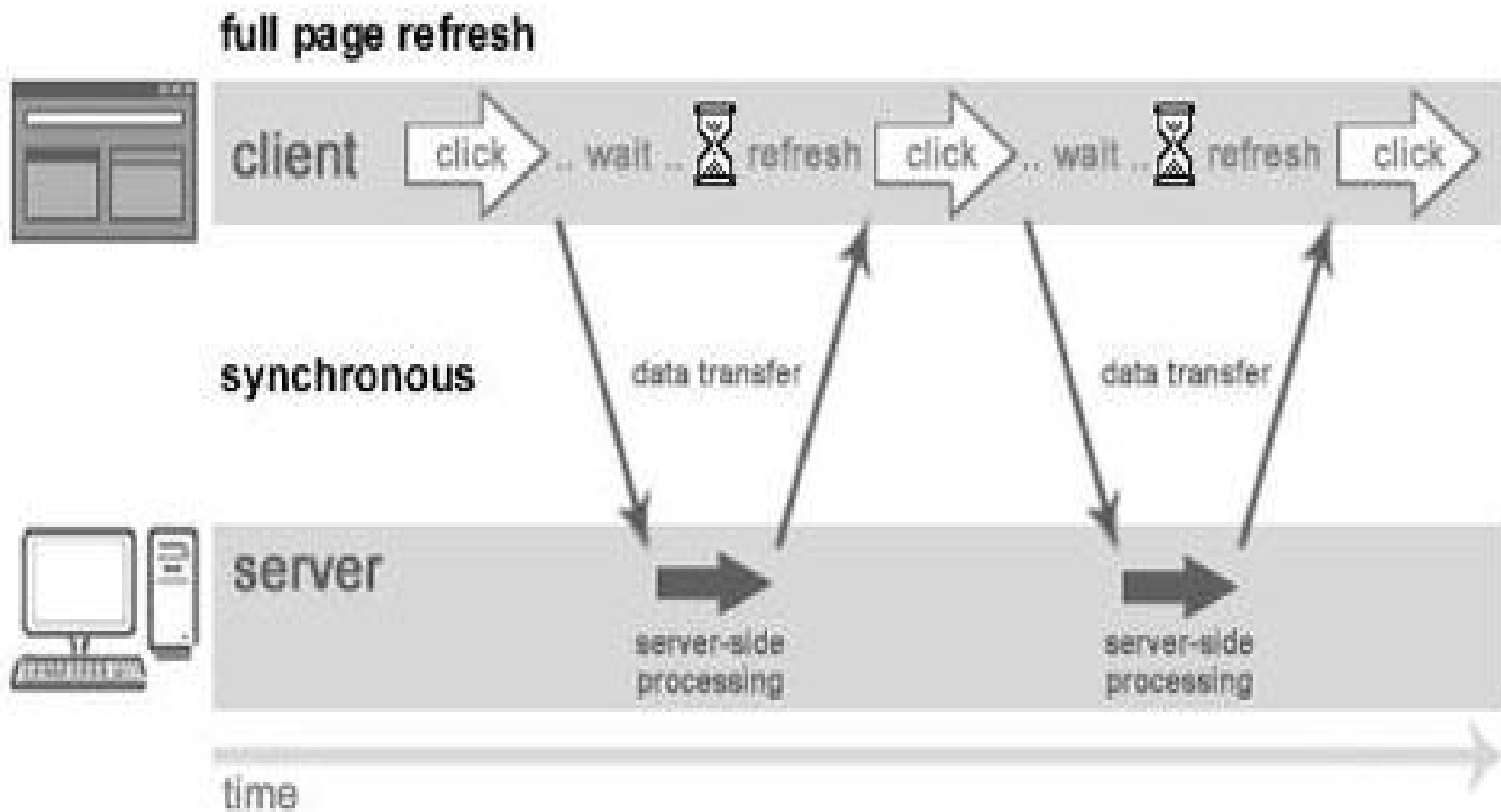
- AJAX là gì?
- Cơ chế truyền thông đồng bộ
- Cơ chế truyền thông bất đồng bộ
- So sánh 2 cơ chế
- Phân tích mã AJAX
- Demo
- Hướng dẫn thực hành

AJAX là gì?

- Asynchronous JavaScript and XML
 - Kỹ thuật tương tác với server theo cơ chế bất đồng bộ
- Nền tảng của AJAX
 - HTML
 - CSS
 - JavaScript (chủ chốt)
 - XML

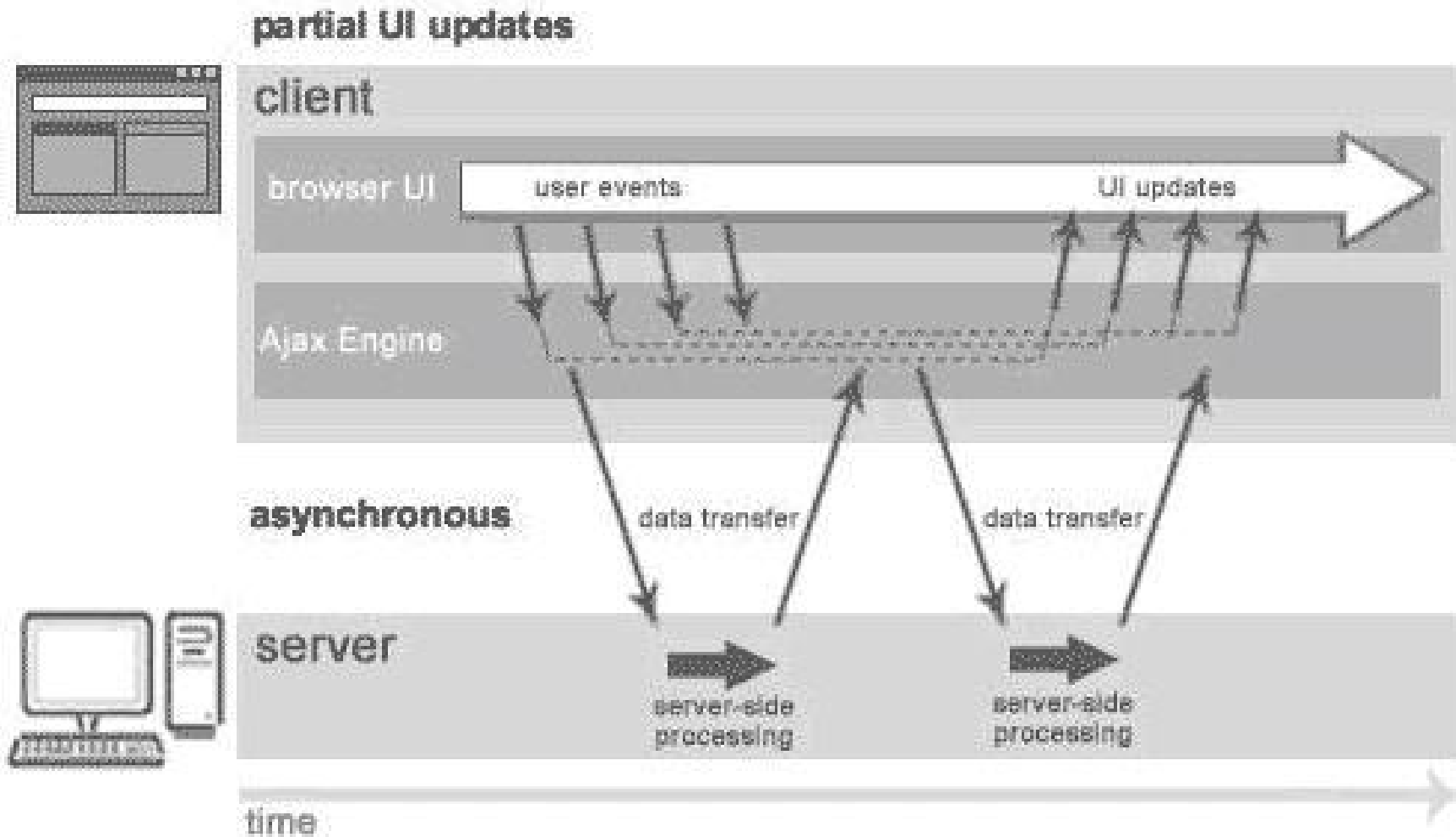
Cơ chế synchronization

4



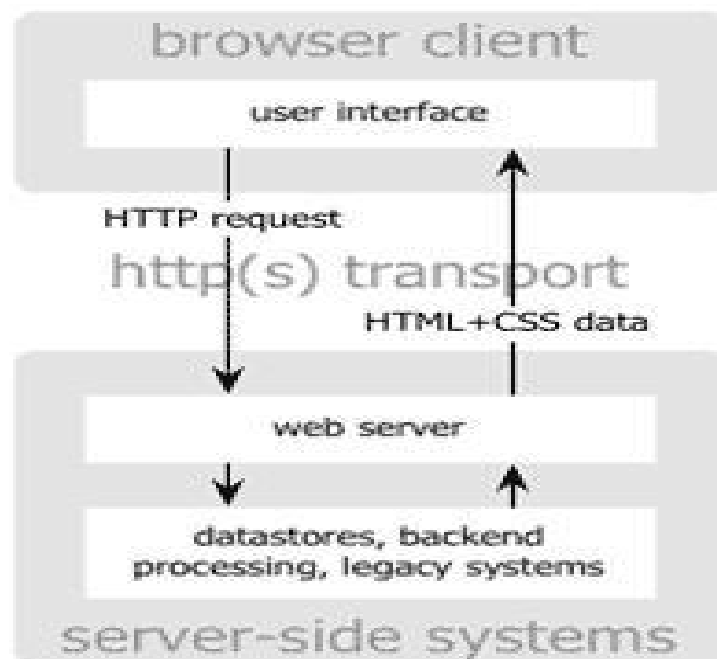
Cơ chế asynchronization

5

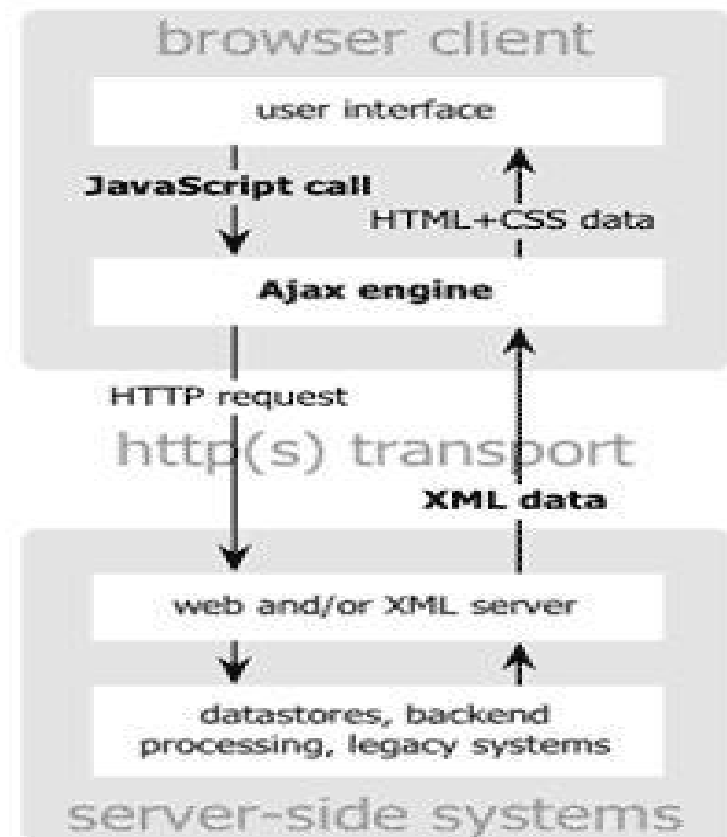


So sánh sync và async

6



classic
web application model



Ajax
web application model

XMLHttpRequest

- Là một đối tượng cho phép **tạo các requests và nhận response theo cơ chế bất đồng bộ** (bằng mã javascript) mà không đòi hỏi thao tác submit của người dùng.
- XMLHttpRequest được cài đặt bởi Microsoft năm 1999 như một ActiveX trong IE và trở thành nhân tố chuẩn cho mọi trình duyệt.

Hoạt động của XMLHttpRequest

8

1. Tạo một thể hiện (instance) của đối tượng XMLHttpRequest, giả sử là Obj
2. Sử dụng đối tượng Obj để tạo một lời gọi (call) bất đồng bộ tới server và **định nghĩa một hàm “gọi làm” – callback**. Hàm này sẽ được thực thi tự động khi nhận được response từ server.
3. Đọc response của server trong hàm callback.
4. Cập nhật lại trang sử dụng các dữ liệu đã nhận được.
5. Quay lại bước 2.

Tạo XMLHttpRequest

- XMLHttpRequest được cài đặt khác nhau trên các trình duyệt.
- Đối với IE:
 - ▣ `Obj = new ActiveXObject("Microsoft.XMLHttp");`
- Đối với các trình duyệt khác: FF, Chrome, Opera...
 - ▣ `Obj = new XMLHttpRequest();`

Các phương thức của XMLHttpRequest

- **open**(method, server, async): đặt các tham số cho request
 - ▣ **Method**: Quy định hình thức vận chuyển dữ liệu đến server. Có 2 hình thức: POST/GET
 - ▣ **Server**: Trang web nhận và xử lý tham số sau đó trả kết quả về client thông qua thuộc tính responseText.
 - ▣ **Async**: Qui định cơ chế truyền thông (**true**: bất đồng bộ, **false**: đồng bộ)
- **send**: gửi request đến server

GET/POST

□ GET:

- `Obj.open("GET", "demo.aspx?id=a&cat=b", true);`
- `Obj.onreadystatechange = ten_ham_callback;`
- `Obj.send(null);`

□ POST:

- `Obj.open("POST", "demo.aspx", true);`
- `Obj.onreadystatechange =
handleRequestStateChange;`
- `Obj.send("id=a&cat=b");`

Các thuộc tính của XMLHttpRequest

<code>readyState</code>	Returns the status of the request: 0 = uninitialized 1 = loading 2 = loaded 3 = interactive 4 = complete
<code>status</code>	Returns the status code of the request.
<code>statusText</code>	Returns the status message of the request.
<code>responseText</code>	Returns the server response as a string.

```
<script language="javascript" type="text/javascript">
    var obj = null;
    function taoxmlhttprequest() {
        if (window.ActiveXObject) //nếu là IE
            obj = new ActiveXObject("Microsoft.XMLHTTP");
        else if (window.XMLHttpRequest) //dành cho trình duyệt còn lại
            obj = new XMLHttpRequest();
    }
    function laydulieu() {
        if (obj) //đã khai báo rồi
        {
            obj.open("GET", "LayNgayGio.aspx", true);
            obj.onreadystatechange = hamcapnhat;
            obj.send(null);
        }
    }

    function hamcapnhat() {
        if (obj.readyState == 4) {
            document.write(obj.responseText);
        }
    }
}
</script>
```

```
<body onload="taoxmlhttprequest();">
    <form id="form1" runat="server">
        <input id="Button3" type="button" value="button"
            onclick="laydulieu();" />
    </form>
</body>
```

14

Ajax Extension

- ScriptManager
- Timer
- UpdatePanel
- UpdateProgress