# Open Source Software Project Development
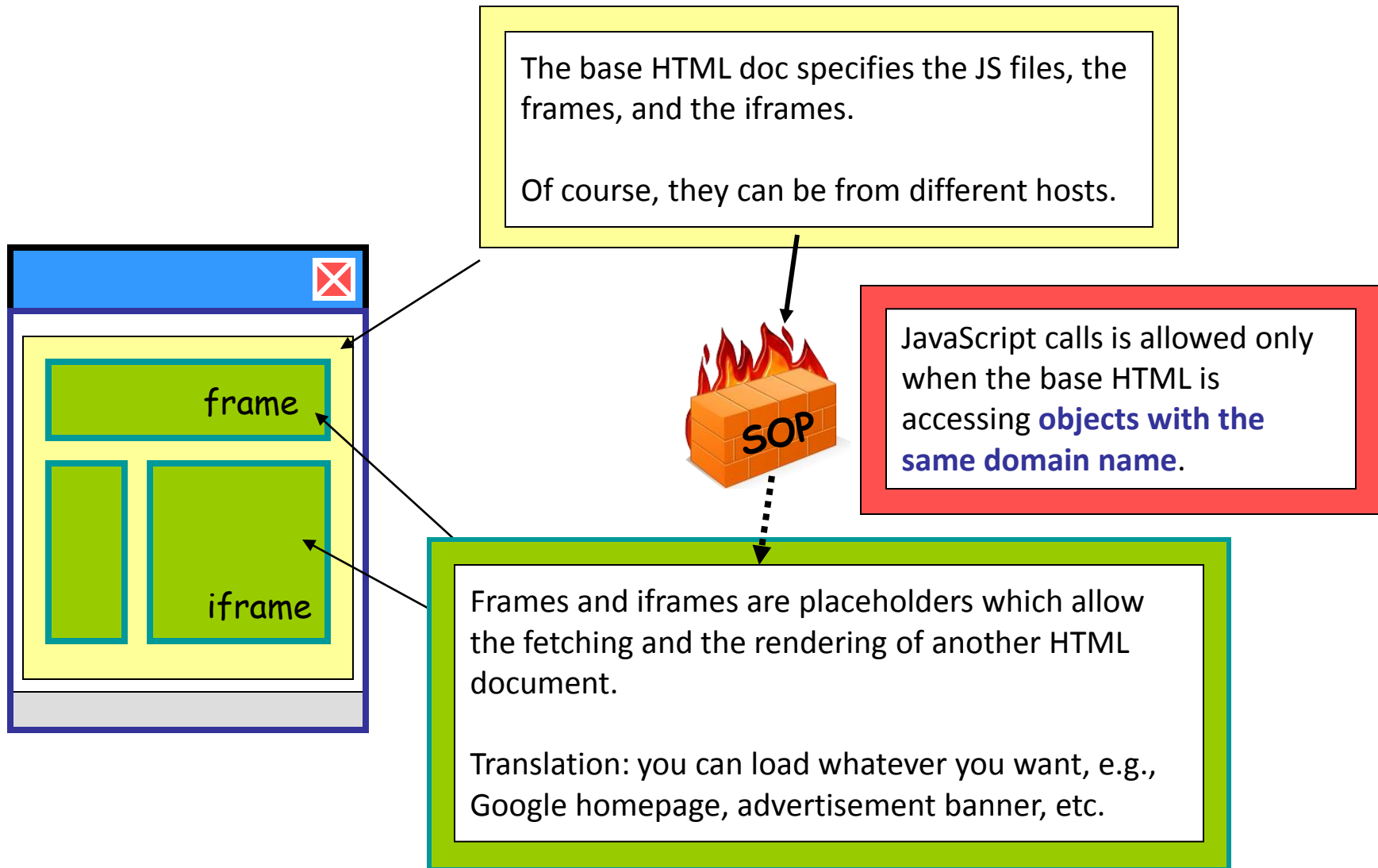
Dr. T.Y. Wong

## Week 10

## **Experiencing Mashup and JSONP**

*- a winning battle against the same origin policy (SOP)*
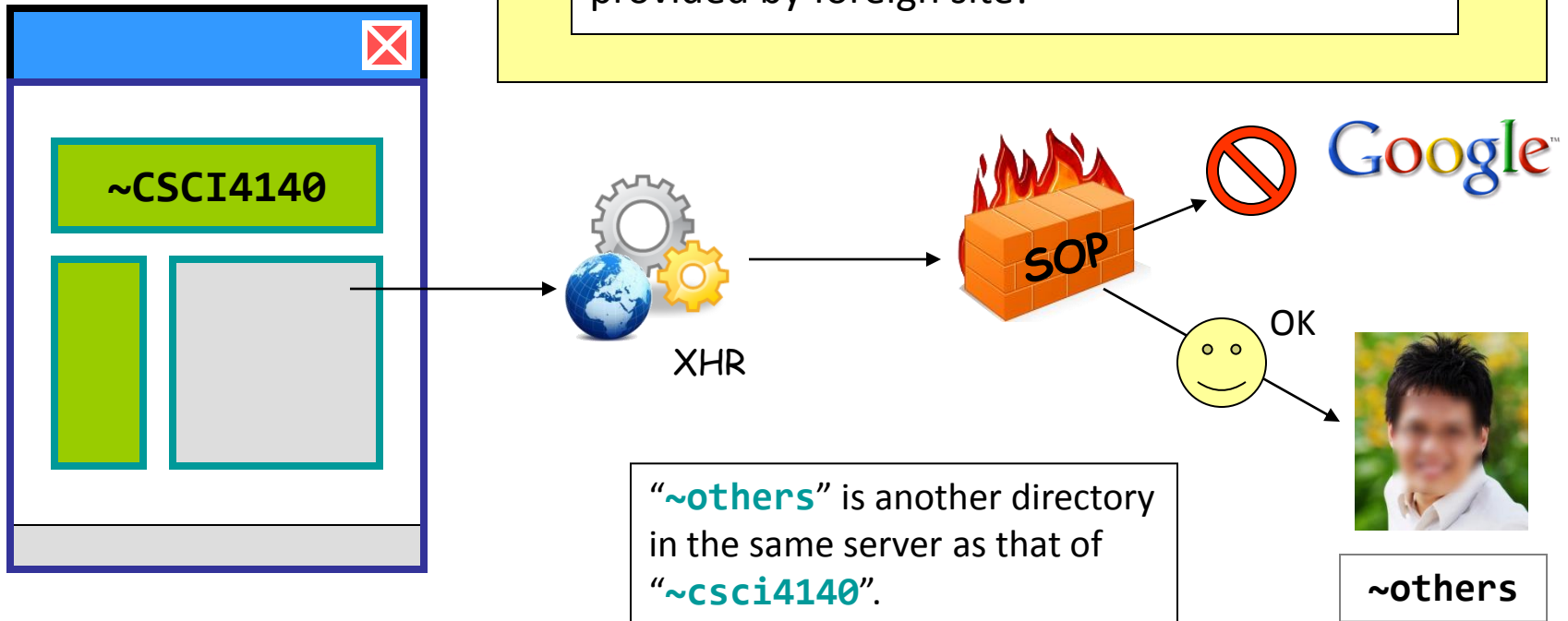
# What is single origin policy (SOP)?

The base HTML doc specifies the JS files, the frames, and the iframes.

Of course, they can be from different hosts.

**frame**

**iframe**

**SOP**

JavaScript calls is allowed only when the base HTML is accessing **objects with the same domain name**.

Frames and iframes are placeholders which allow the fetching and the rendering of another HTML document.

Translation: you can load whatever you want, e.g., Google homepage, advertisement banner, etc.

# What is single origin policy (SOP)?

**What does the SOP control imply?**

You cannot use AJAX to access services provided by foreign site!

~CSCI4140

XHR

SOP

Google

OK

"**~others**" is another directory in the same server as that of "**~csci4140**".

**~others**

# What is a Mashup?

- A mashup is a website or application that combines contents from more than one source.

A mixed search
hello
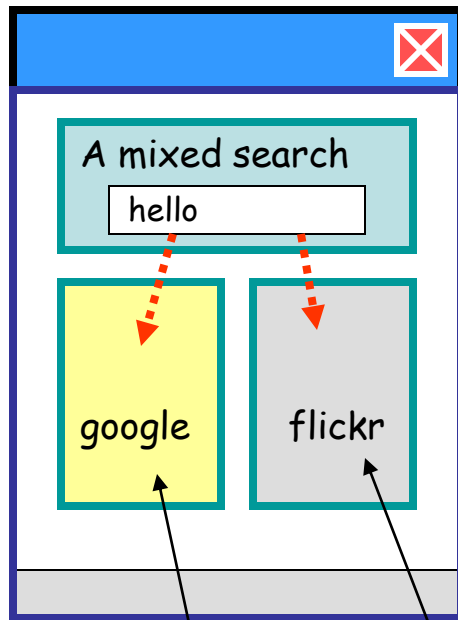
google    flickr

**Define our goal**

Write a sexy app that perform a mixed search on both Google and Flickr at the same time!
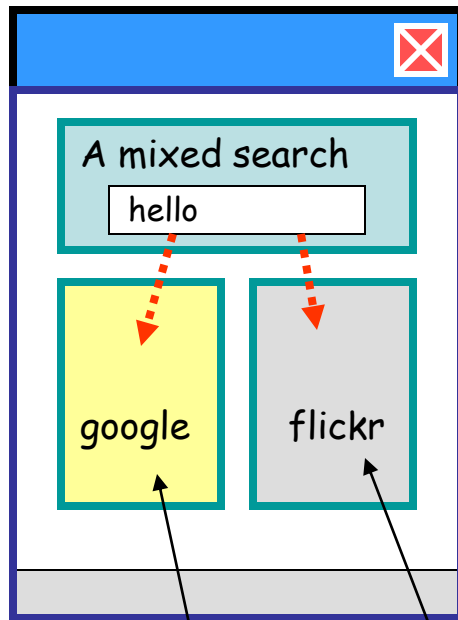
**What can we do if both are divs?**

No way!

The divs cannot load nor display contents from other domains using AJAX....

# What is a Mashup?

- A mashup is a website or application that combines contents from more than one source.

**A mixed search**

hello

google | flickr

**Define our goal**

Write a sexy app that perform a mixed search on both Google and Flickr at the same time!
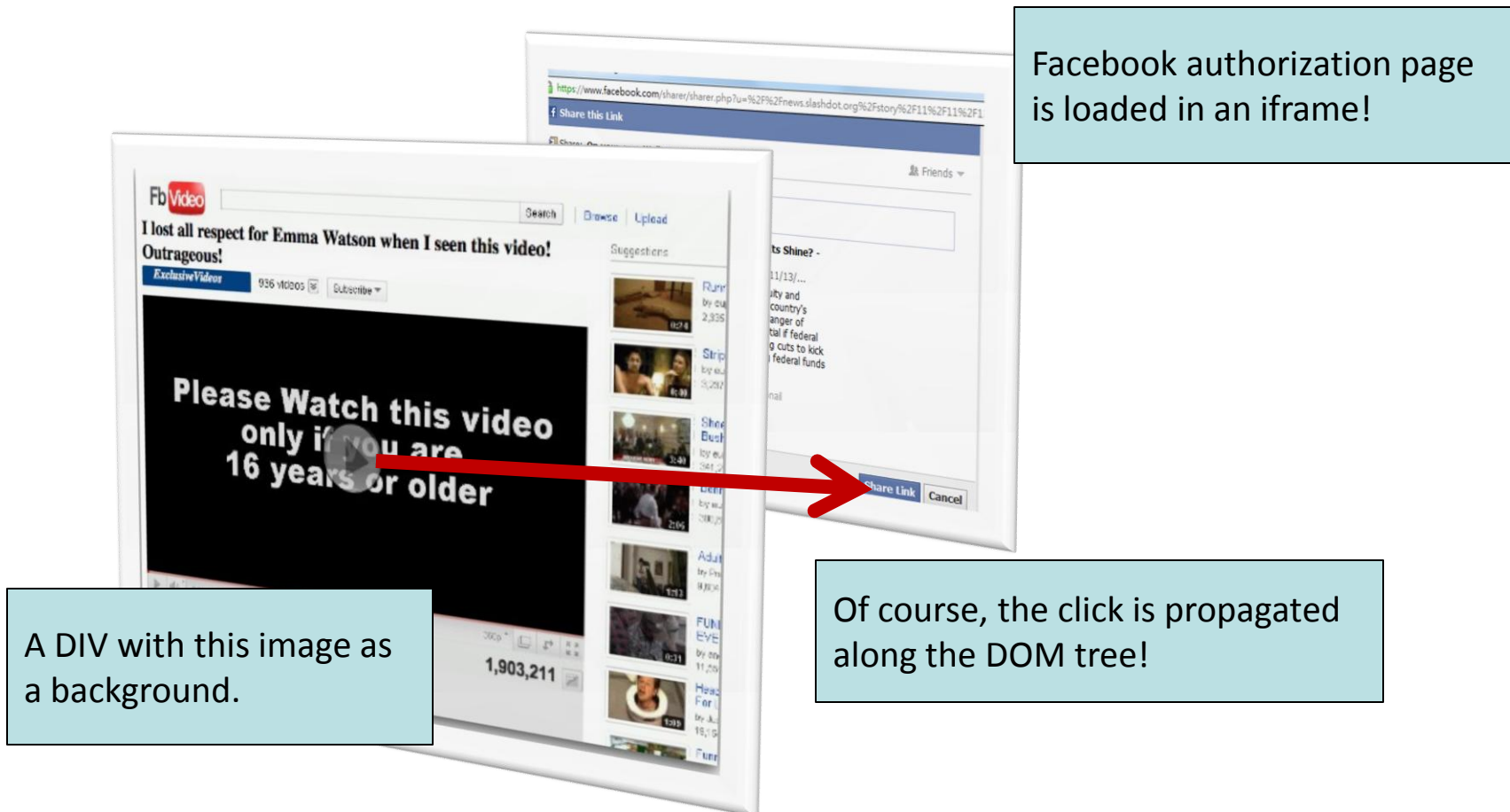
**What can we do if both are iframes?**

Use DOM manipulation to change the "**src**" fields of the iframes.

No way!!  Because of **X-FRAME-OPTIONS**.

`See "iframe_*.html"`

# Sidetrack – Clickjacking & X-FRAME-OPTION

- Clickjacking cheats your clicks!



Facebook authorization page is loaded in an iframe!

A DIV with this image as a background.

Of course, the click is propagated along the DOM tree!

See "clickjack.html"

# Sidetrack – Clickjacking & X-FRAME-OPTIONS

- This attack cheats your clicks!



**Defense.**

-The server adds a header called "**X-FRAME-OPTIONS**" to the response header.
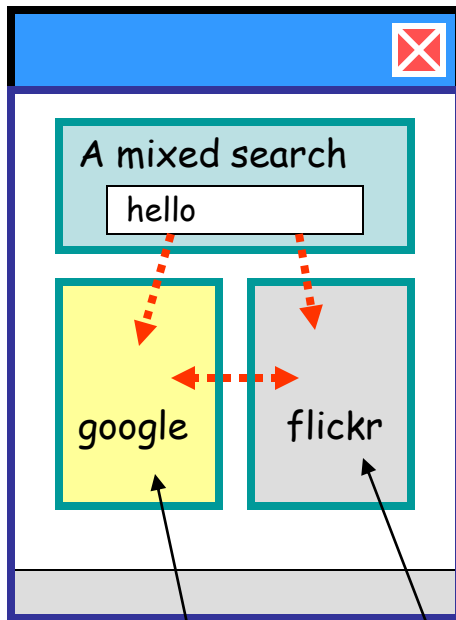
-The browser reacts to the value of the header.

If the value is **SAMEORIGIN**, the page can then be displayed in a frame on the same origin.

If the value is **DENY**, the page cannot be displayed in any frames.

# What is a Mashup?

- A mashup is a website or application that combines contents from more than one source.

**Define our goal**

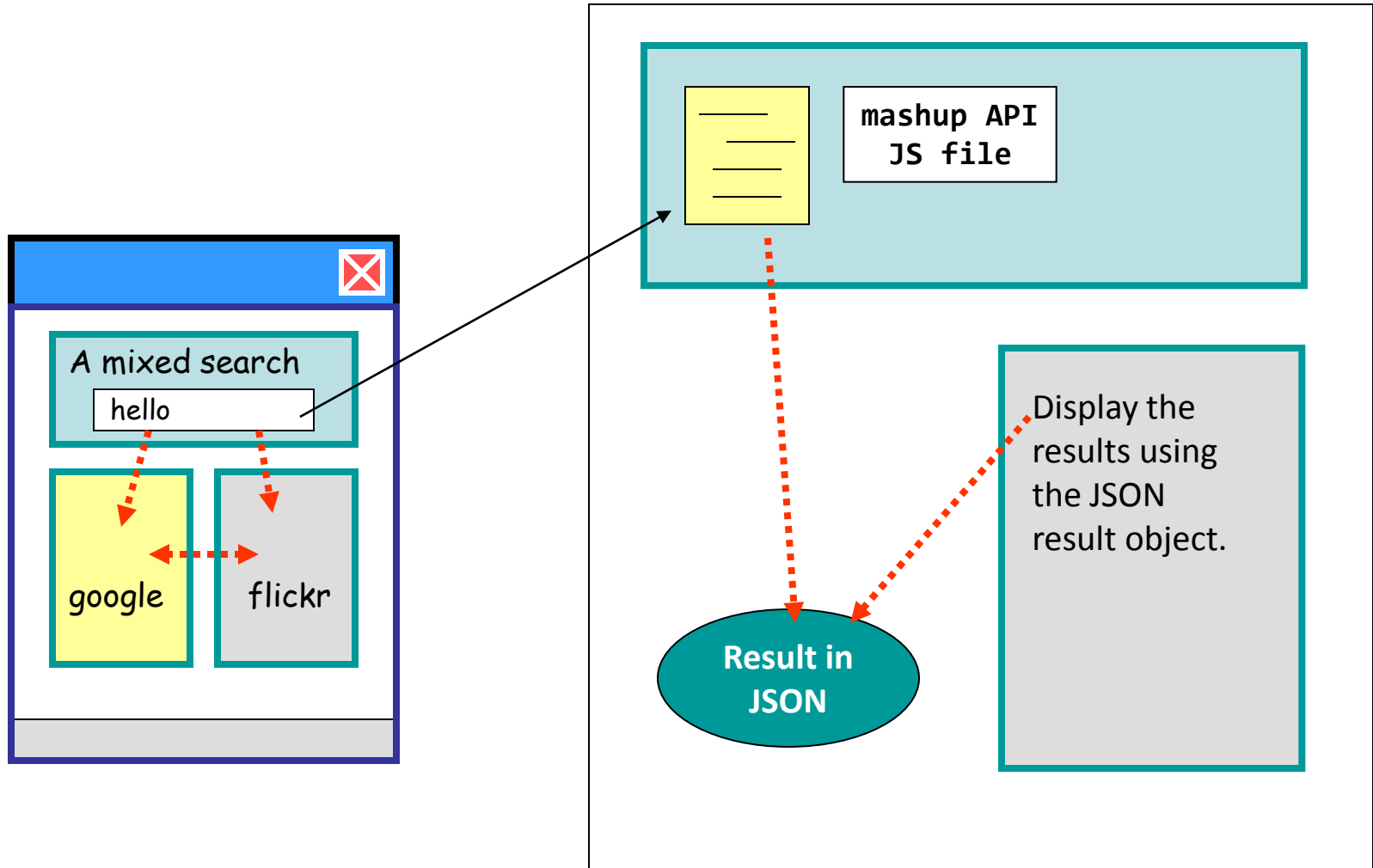> Write a sexy app that perform a mixed search on both Google and Flickr at the same time!

A mixed search
hello

google    flickr
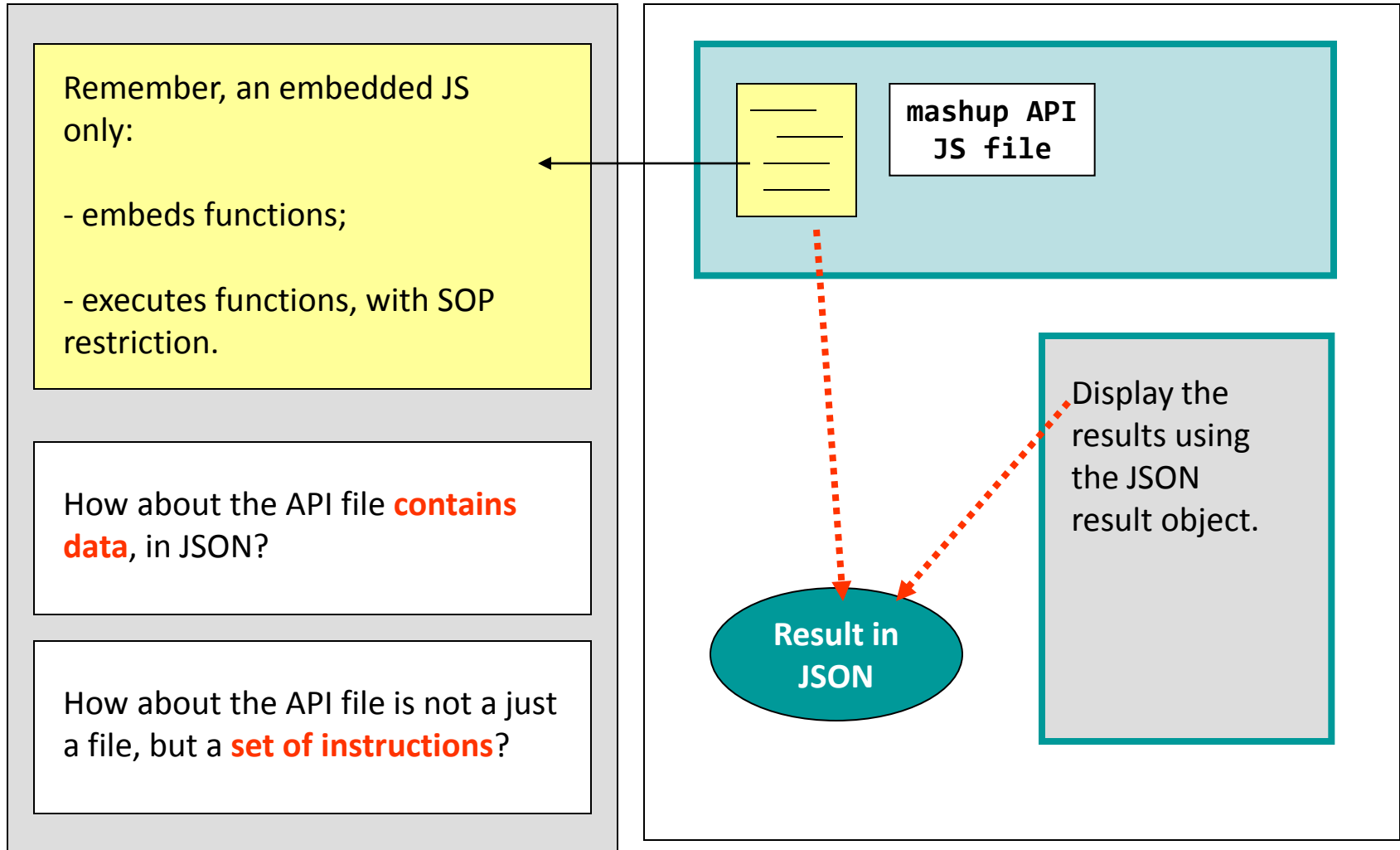
**What if they need to communicate?**

> No way again...

> So, does it mean that mashup apps are born to be restrictive?
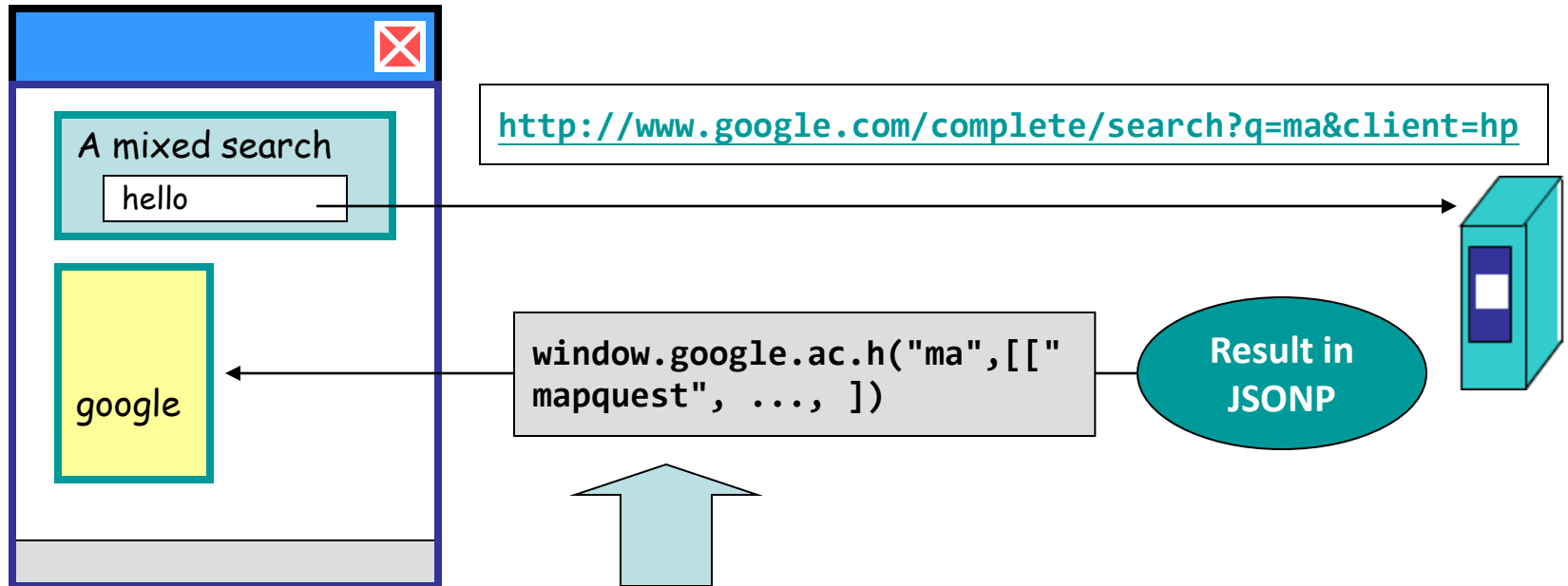
See "mashup_search_*.html", "google_complete.html"

# Mashup API design…

# Mashup API design...

Remember, an embedded JS only:

- embeds functions;

- executes functions, with SOP restriction.

How about the API file **contains data**, in JSON?

How about the API file is not a just a file, but a **set of instructions**?

```
mashup API
JS file
```

Display the results using the JSON result object.

**Result in JSON**

# Mashup API design...with JSONP

A mixed search

hello

`http://www.google.com/complete/search?q=ma&client=hp`

```
window.google.ac.h("ma",[["
mapquest", ..., ])
```
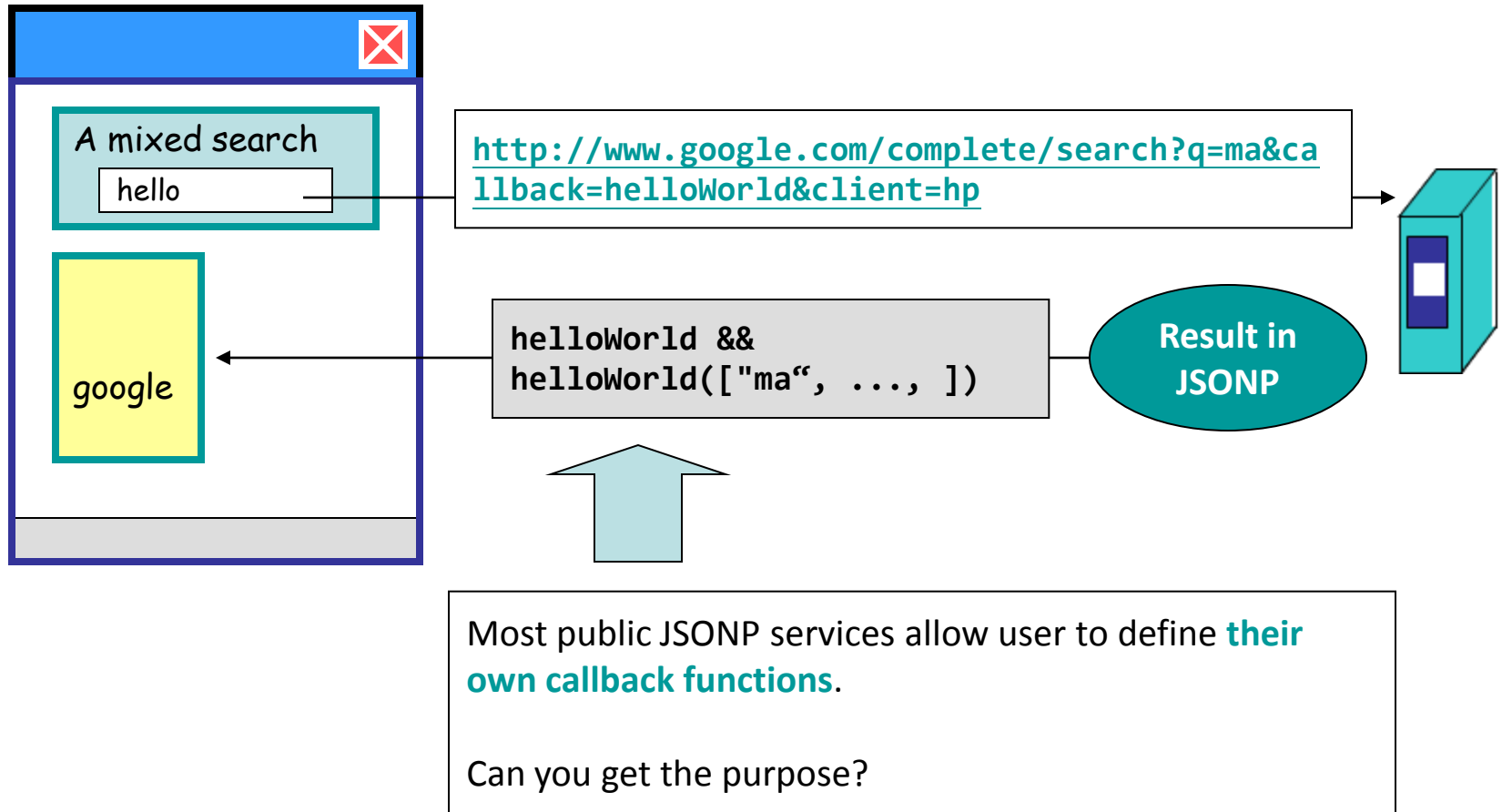
google

**Result in JSONP**

This technique is known as **JSON with padding** (or JSONP).
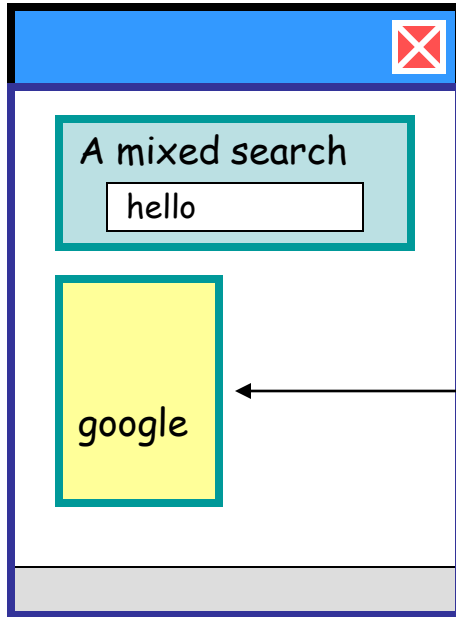
Google's "index.html" expects the URL is called using its "**index.html**". Then, there should exist a function **window.google.ac.h()**.

Then, Google's server pads the function name together with the result in terms of JSON!

# Mashup API design…with JSONP

A mixed search

`hello`

google

http://www.google.com/complete/search?q=ma&callback=helloWorld&client=hp

```
helloWorld &&
helloWorld(["ma", ..., ])
```

**Result in JSONP**

Most public JSONP services allow user to define **their own callback functions**.

Can you get the purpose?

# JSONP example

A mixed search

hello

google

```html
<html>
<script>
function helloWorld(jsonObj) {
    var results = jsonObj[1];
    for(var i = 0; i < results.length; i++) {
        var p = document.createElement("p");
        p.innerHTML = results[i][0];

        var array = document.getElementsByTagName("body");
            array[0].appendChild(p);
    }
}
</script>

<body>
<script src=
  "http://www.google.com/complete/search?q=ma&callback=helloWorld&
client=hp">
</script>
</body>
```
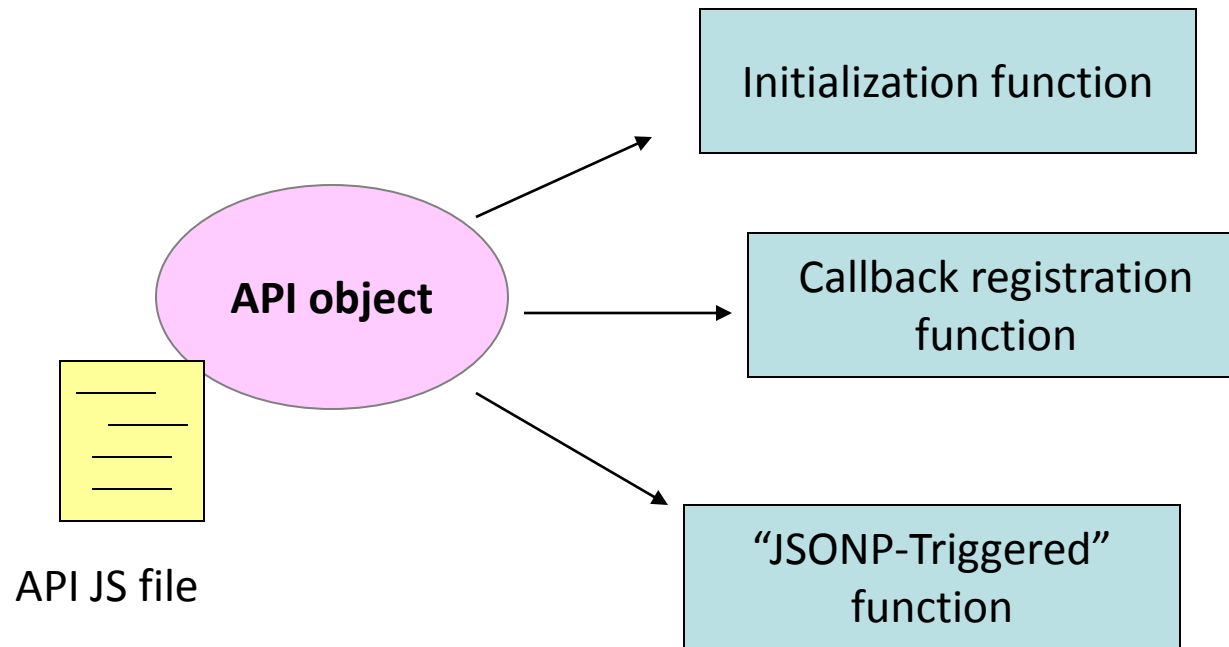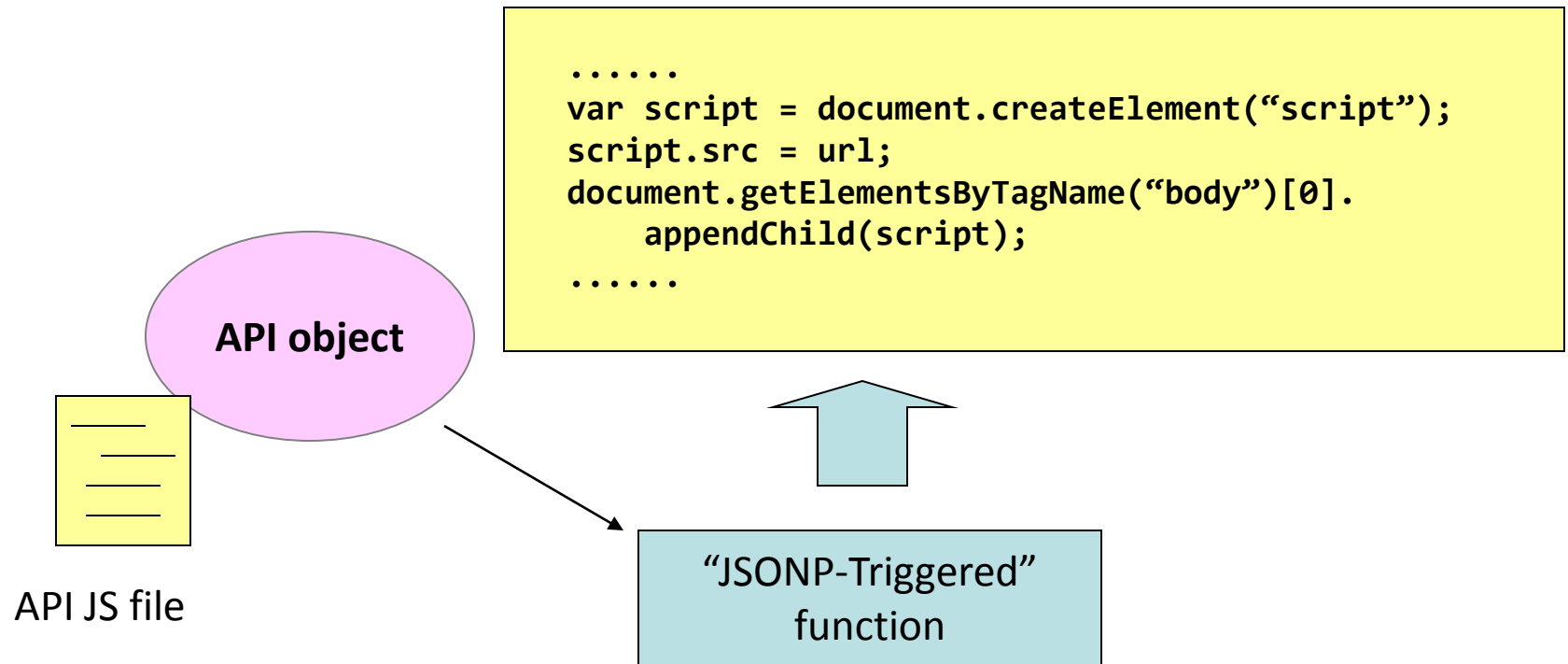
using
JSONP

**See "jsonp.html"**

# Write an API using JSONP…

- The previous example is not flexible enough…



API JS file
API object
Initialization function
Callback registration function
"JSONP-Triggered" function

# Write an API using JSONP…

- The previous example is not flexible enough…

```
......
var script = document.createElement("script");
script.src = url;
document.getElementsByTagName("body")[0].
    appendChild(script);
......
```

**API object**

API JS file

"JSONP-Triggered" function

# Public APIs...References

- Google search:
  - http://code.google.com/apis/loader/

- Google map:
  - http://code.google.com/apis/maps/

- Flickr:
  - http://www.flickr.com/services/api/