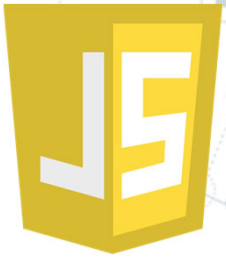
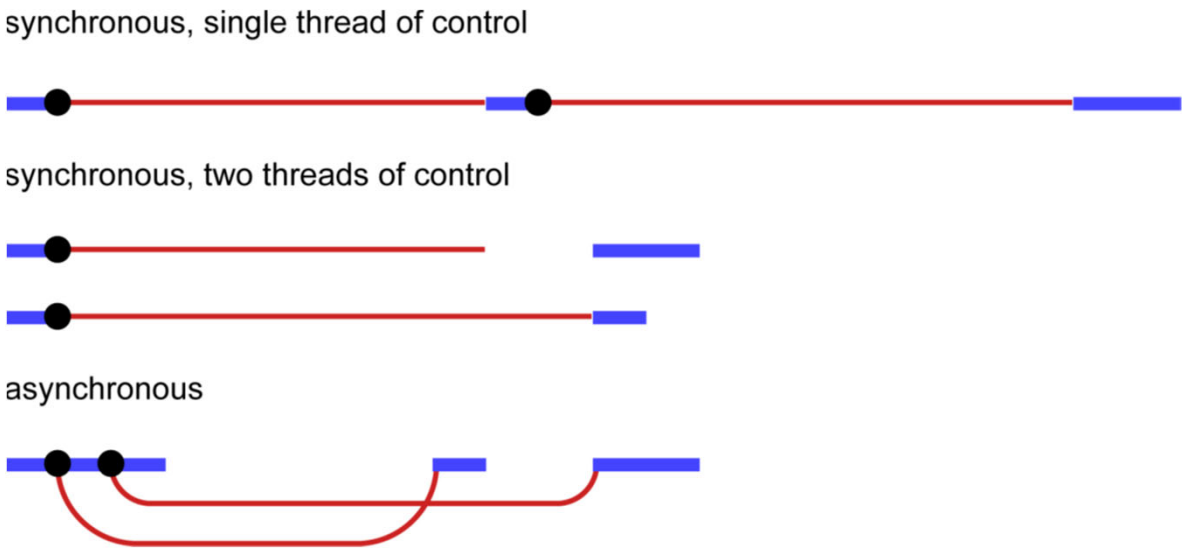
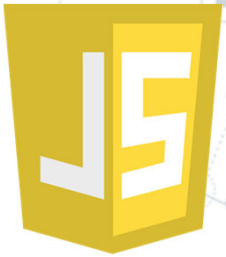
Javascript



JavaScript is always synchronous and single-threaded. If you're executing a JavaScript block of code on a page then no other JavaScript on that page will currently be executed.



Javascript – Callback and Promise



One approach to asynchronous programming is to make functions that perform a slow action take an extra argument, a *callback function*. The action is started, and when it finishes, the callback function is called with the result.

```
setTimeout(() => console.log("Tick"), 500);
```

A *promise* is an asynchronous action that may complete at some point and produce a value. It is able to notify anyone who is interested when its value is available.

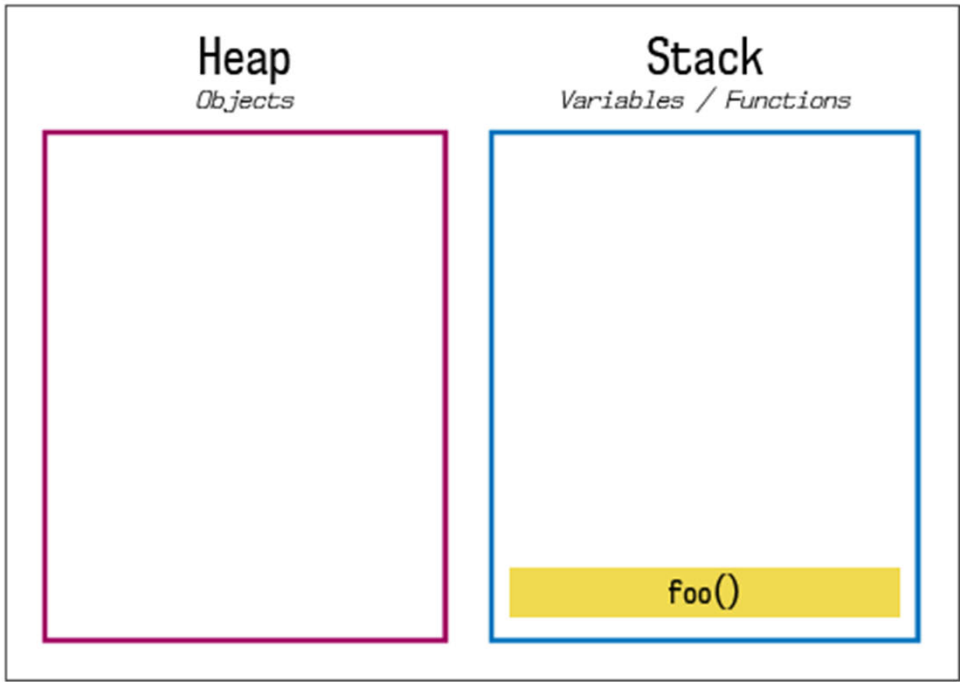
```
let fifteen = Promise.resolve(15);
fifteen.then(value => console.log(`Got ${value}`));
```

Javascript – Callback and Promise

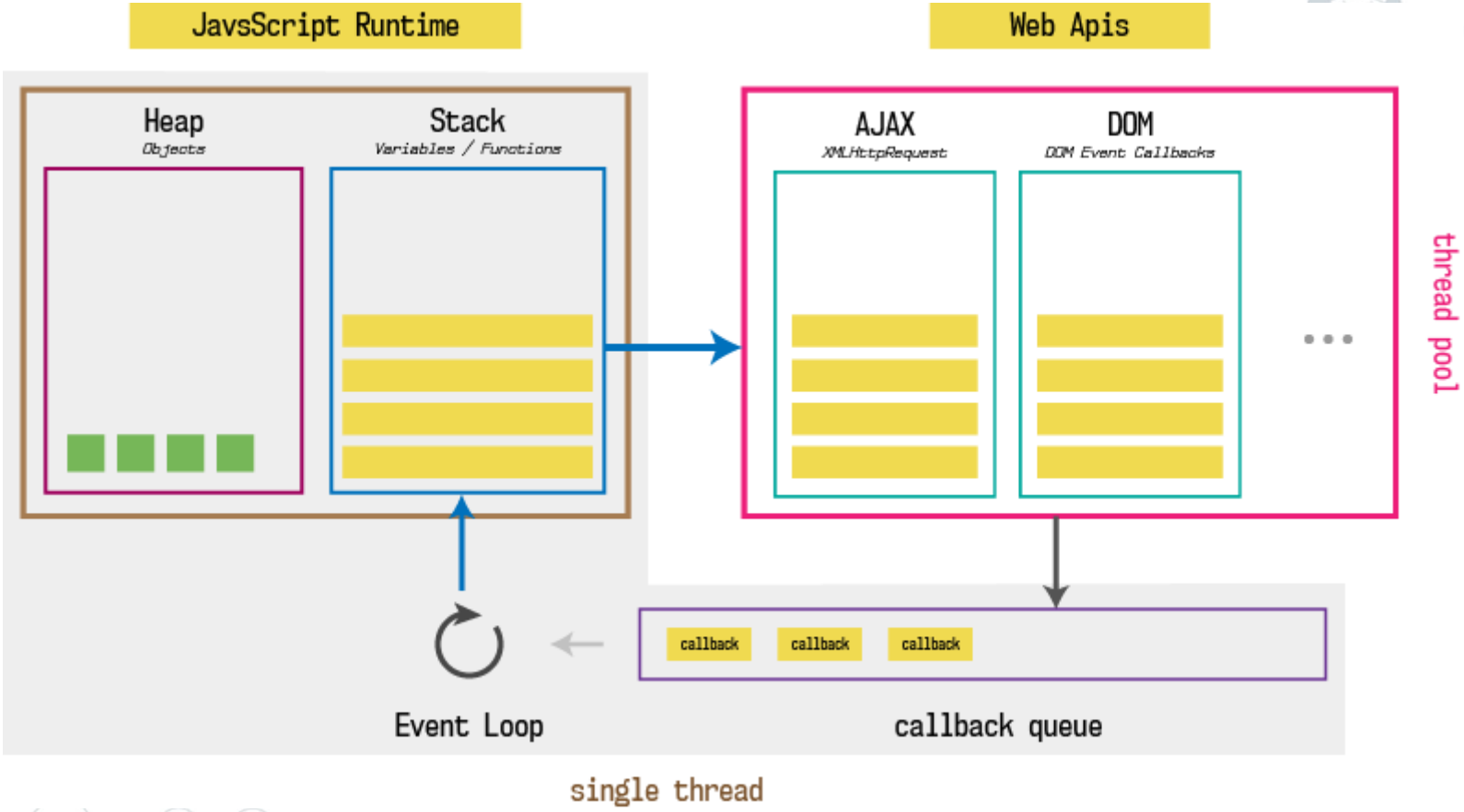
JavsScript Program

```
function baz(){
   console.log('Hello from baz');
}

function bar() {
   baz();
}

function foo() {
   bar();
}

foo();
```

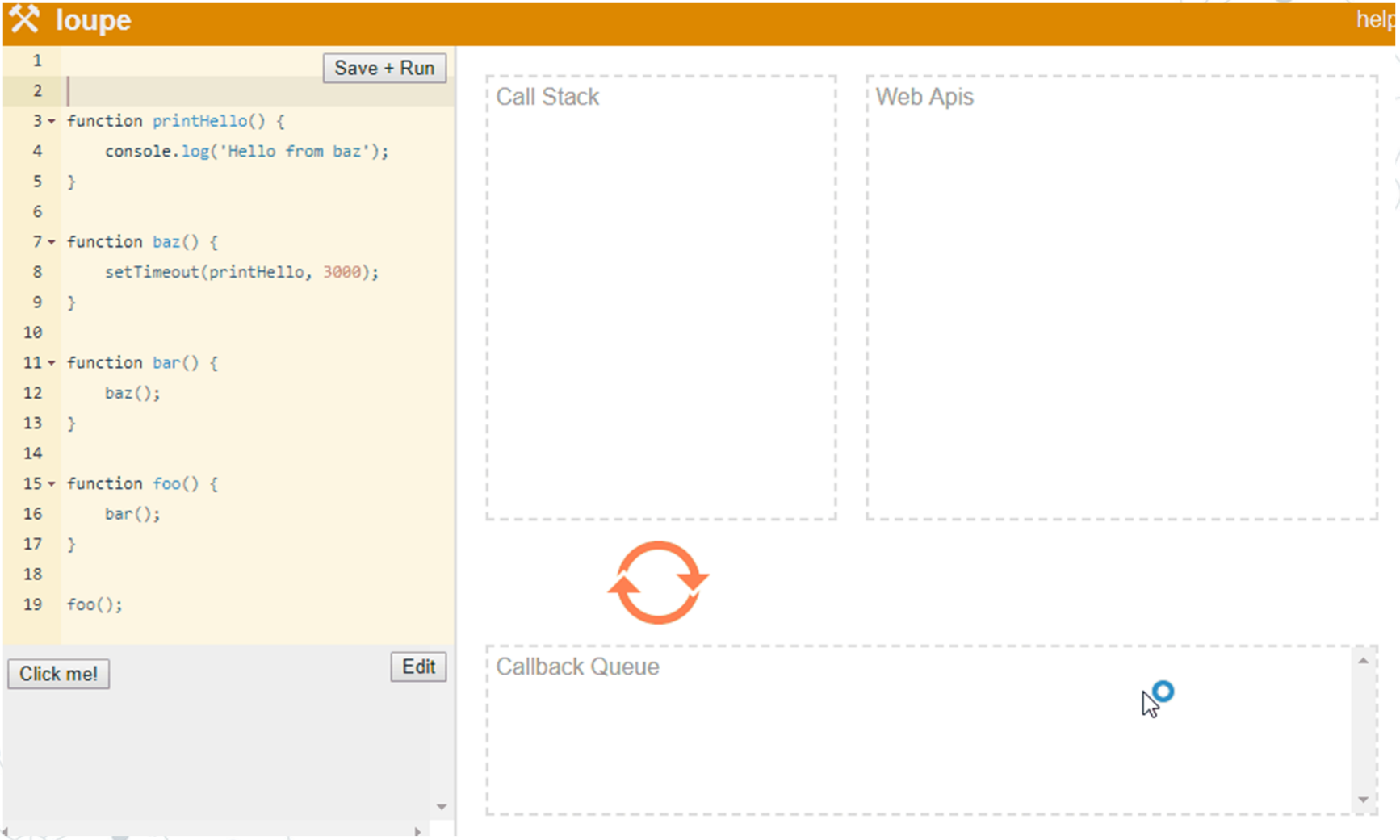
JavsScript Runtime



Javascript – Callback and Promise

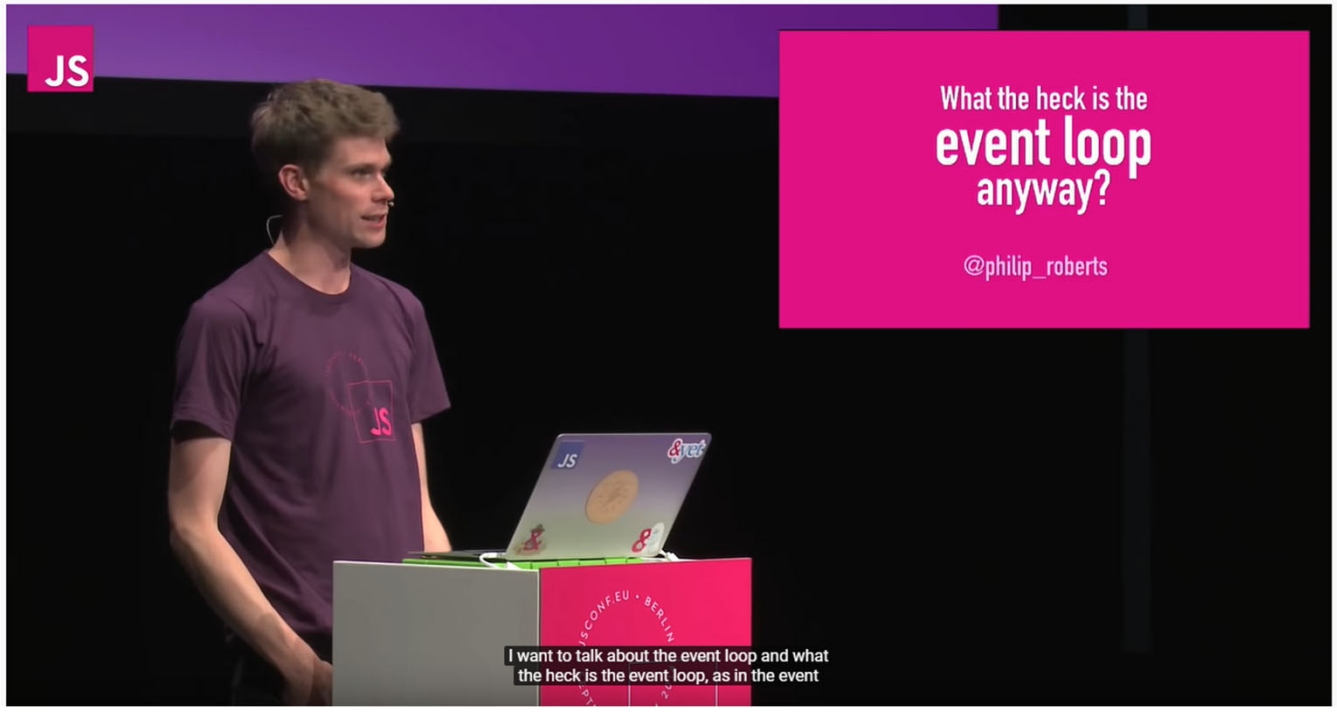


Javascript – Callback and Promise



Javascript – Callback and Promise

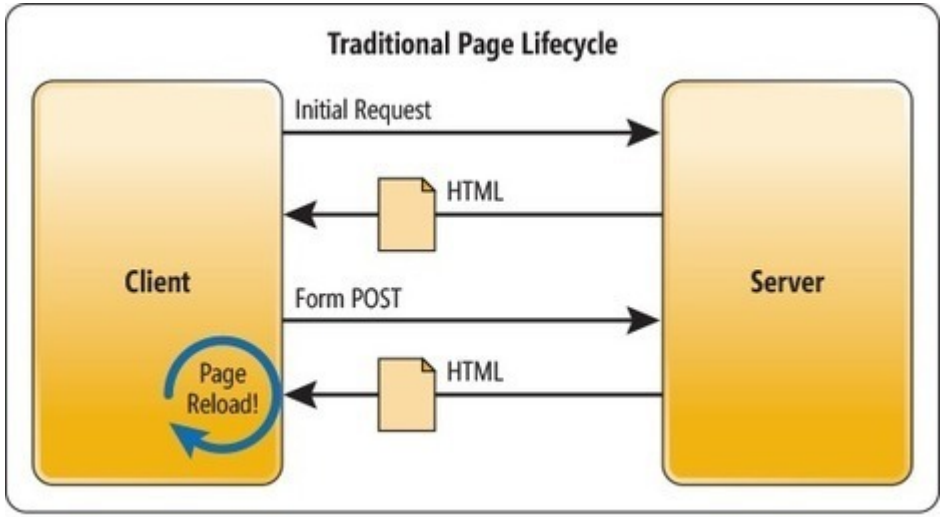
https://www.youtube.com/watch?v=8aGhZQkoFbQ



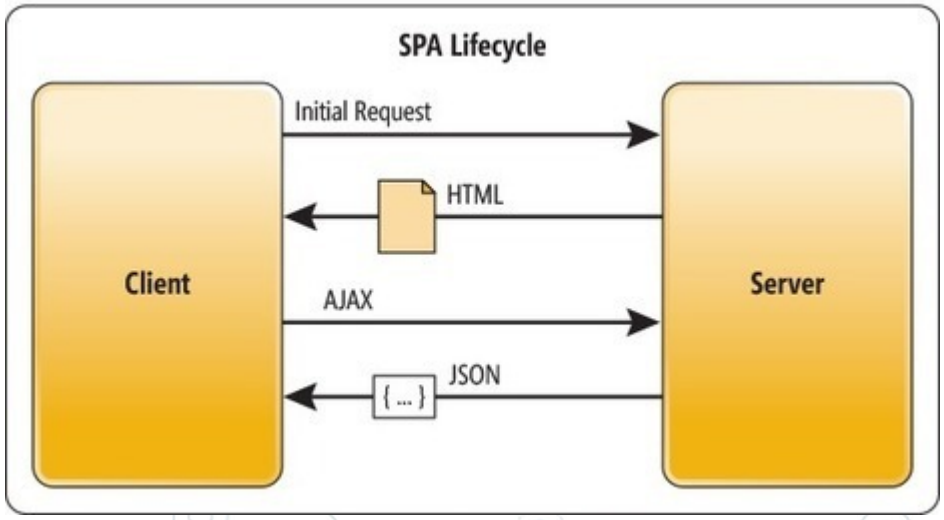
Web e pattern architetturali

Pattern architetturali

Multi-Page Application



Single-Page Application

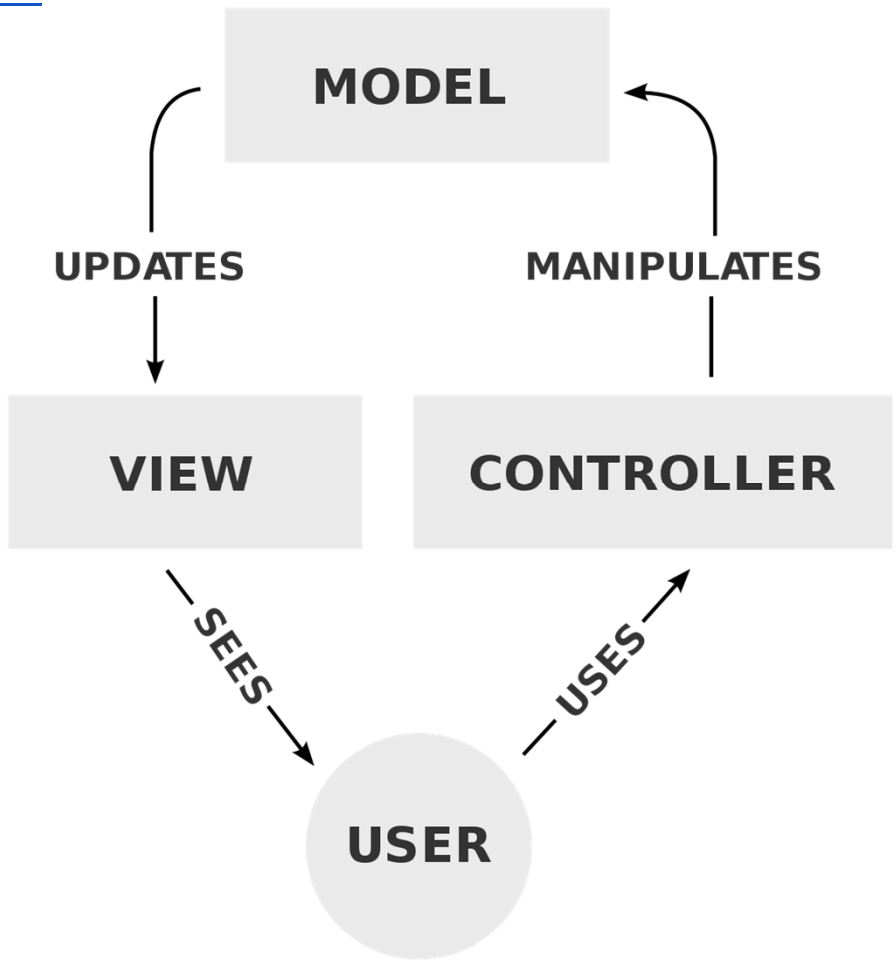


Pattern MVC

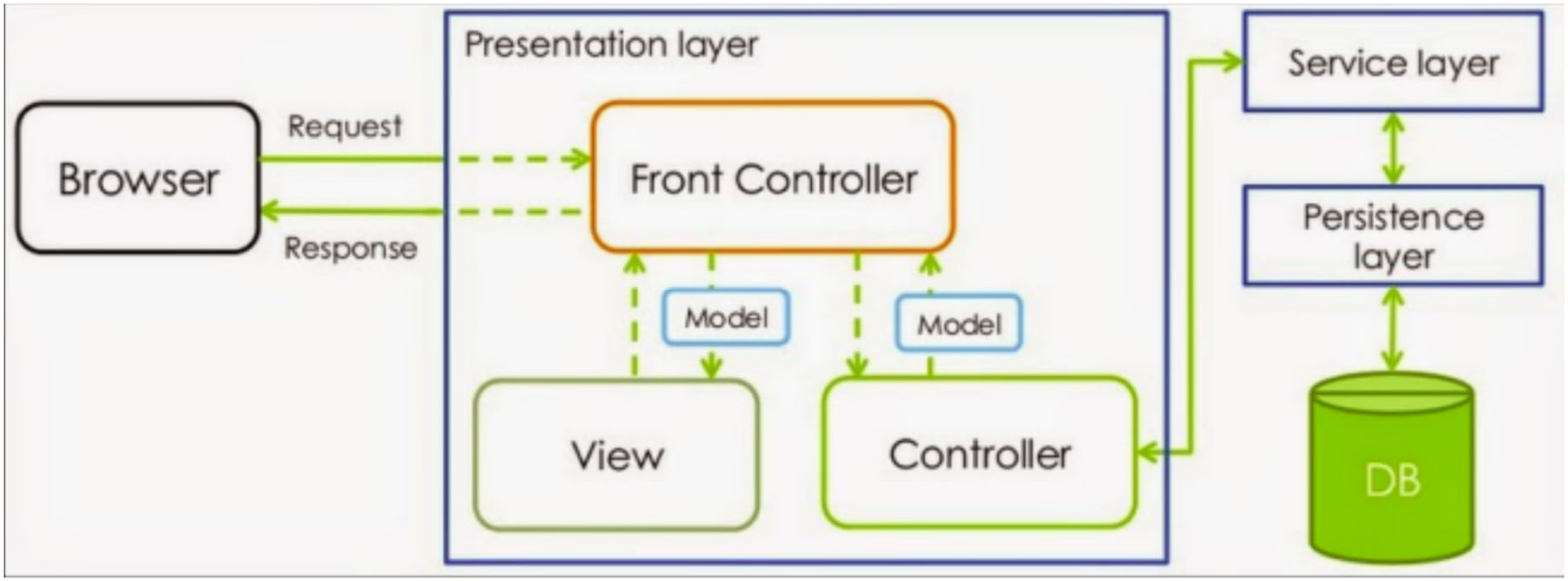
https://it.wikipedia.org/wiki/Model-view-controller

Vantaggi:

1) Disaccoppiare
2) Responsabilità certe
3) View multiple



Architettura generica



The presentation layer is where the data is formatted and presented to the user.

The service layer is wherethe business logic of the application is implemented.

The persistence layer is where the data is simply saved or retrieve.

AJAX – L'inizio delle SPA

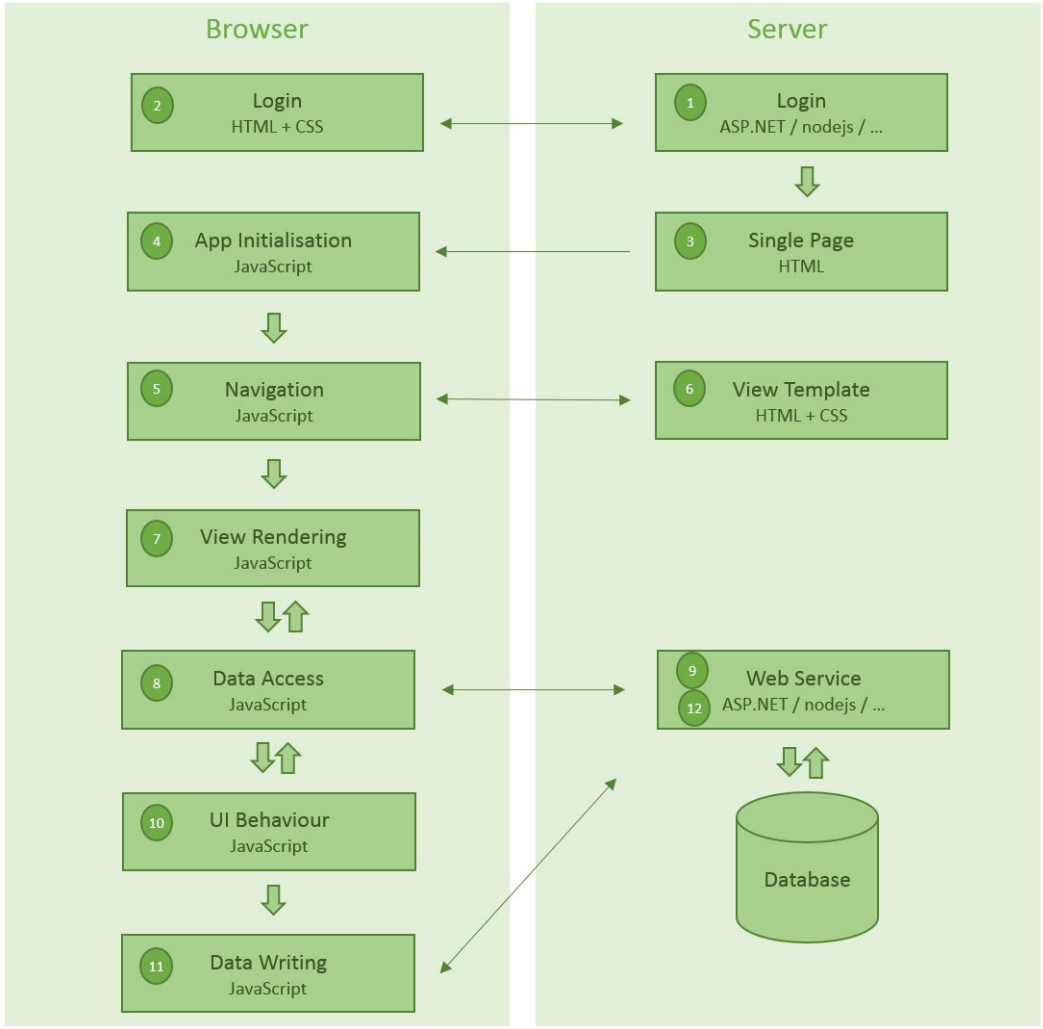
https://embed.plnkr.co/rgh75JGDGuyB4UhBvTYN/

```
<body>
  <h1>Hello <span id="firstname"></span> <span id="lastname"></span>!</h1>
  <button onclick="reload_user();">Reload!</button>
  <p id="loadingtext">LOADING....</p>
</body>

</html>
```

```
$(document).ready(function() {
  reload_user();
});

function reload_user(){
    $("#loadingtext").show();
    $.ajax({
    method: "get",
    url: "https://randomuser.me/api",
    datatype: "json",
    data: { results: 1 },
    success: function(r) {
      $("#firstname").text(r.results[0].name.first);
      $("#lastname").text(r.results[0].name.last);
    },
    error: function() {
      alert("error");
    },
    complete: function(){
      $("#loadingtext").fadeOut();
    }
  });
}
```

Asynchronous Javascript And XML

Esempio di SPA



Il mondo reale è composto da soluzioni ibride

Frontend

Backend

Approcci ibridi
Mix di soluzioni
Evoluzione continua

Trend:
- PWA vs Mobile
- Low Code
- Serverless
- Static site generators
- MicroService

UI Bakery
https://www.youtube.com/watch?v=xbB3MrEi5bo

Less servers for your Angular app
https://www.youtube.com/watch?v=WEYtDYBkaII

Mastering Chaos - A Netflix Guide to Microservices
https://www.youtube.com/watch?v=CZ3wIuvmHeM

Security

SQL Injection

Cosa è:

SQL injection è una tecnica di *code injection* dove si inietta del codice SQL

```
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "'"

# Execute the SQL statement
database.execute(sql)
```

https://www.acunetix.com/websitesecurity/sql-injection/

Come si combatte?

Semplicemente usando: **prepared statements and parameterized queries**

```
$stmt = $dbConnection->prepare('SELECT * FROM employees WHERE name = ?');
$stmt->bind_param('s', $name);
```

Oppure pulendo tutti gli input:

```
mysqli_real_escape_string ( mysqli $link , string $escapestr ) : string
```

This function is used to create a legal SQL string that you can use in an SQL statement. The given string is encoded to an escaped SQL string, taking into account the current character set of the connection.

```
$unsafe_variable = $_POST["user-input"];
$safe_variable = mysql_real_escape_string($unsafe_variable);
mysql_query("INSERT INTO table (column) VALUES ('" . $safe_variable . "')");
```