REACT WORKSHOP

AN INTRODUCTION

Gustavo Jordão
jordao.f.gustavo@gmail.com
github.com/gjordao

EVODECK

SOFTWARE

Website: https://evodeck.com/

Facebook: https://www.facebook.com/evodeck/

Email: hello@evodeck.com

Slack registration: http://gsslackin.herokuapp.com/

Slack: https://geeksessions.slack.com/

Facebook: https://www.facebook.com/GeekSessionsFaro/

IPBeja

Website: https://www.ipbeja.pt/Paginas/default.aspx

AGENDA

- Introduction talk (≤ 20 mins)
- Live Coding (≤ 40 mins)
- Workshop (≤ 120 mins)

INTRODUCTION TALK

- Before we begin be sure to have installed NodeJS
- If you haven't you can download it from:

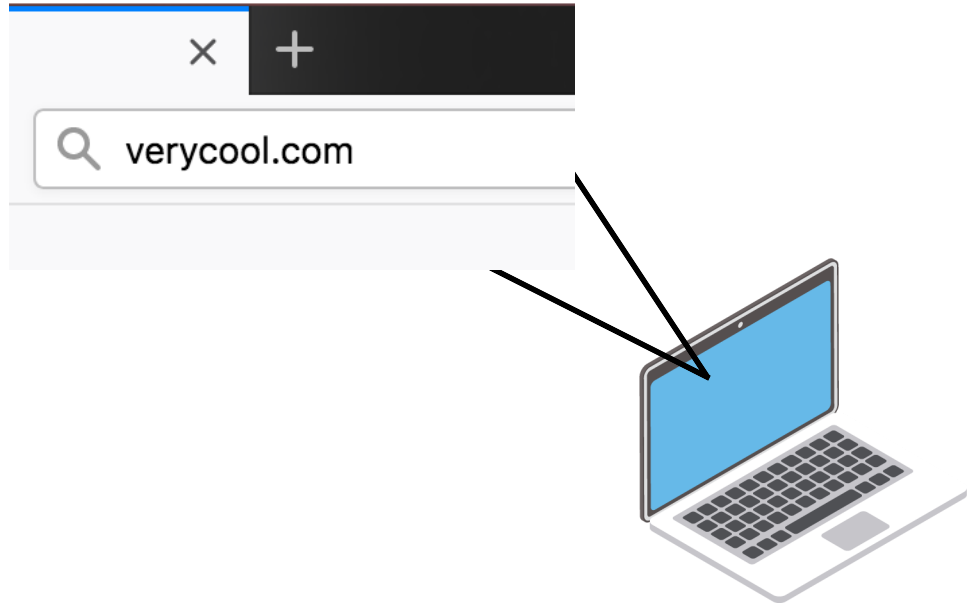
  https://nodejs.org/en/

INTRODUCTION TALK

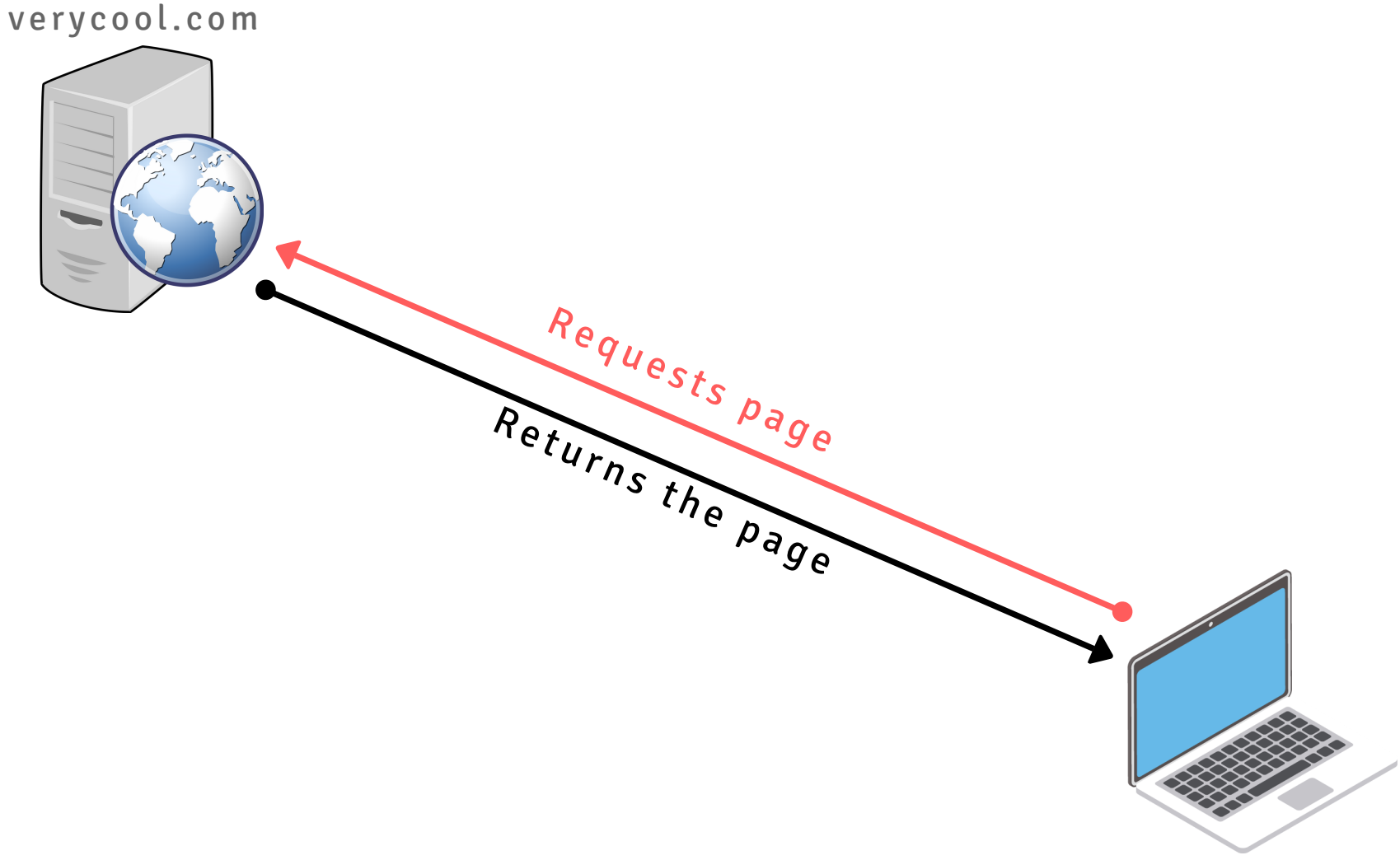
- The web
- Server-side rendering
- Client-side rendering
- APIs
- React

HOW DOES THE WEB WORK TODAY?

verycool.com

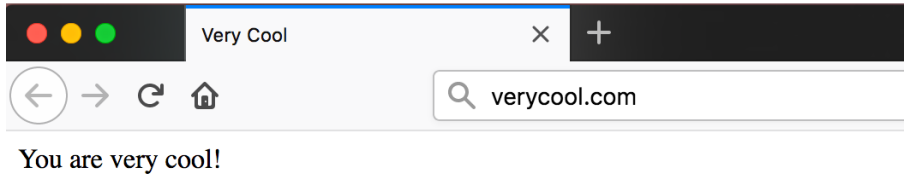


HOW DOES THE WEB WORK TODAY?



HOW DOES THE WEB WORK TODAY?

verycool.com



HOW DOES THE WEB WORK TODAY?

Conclusion

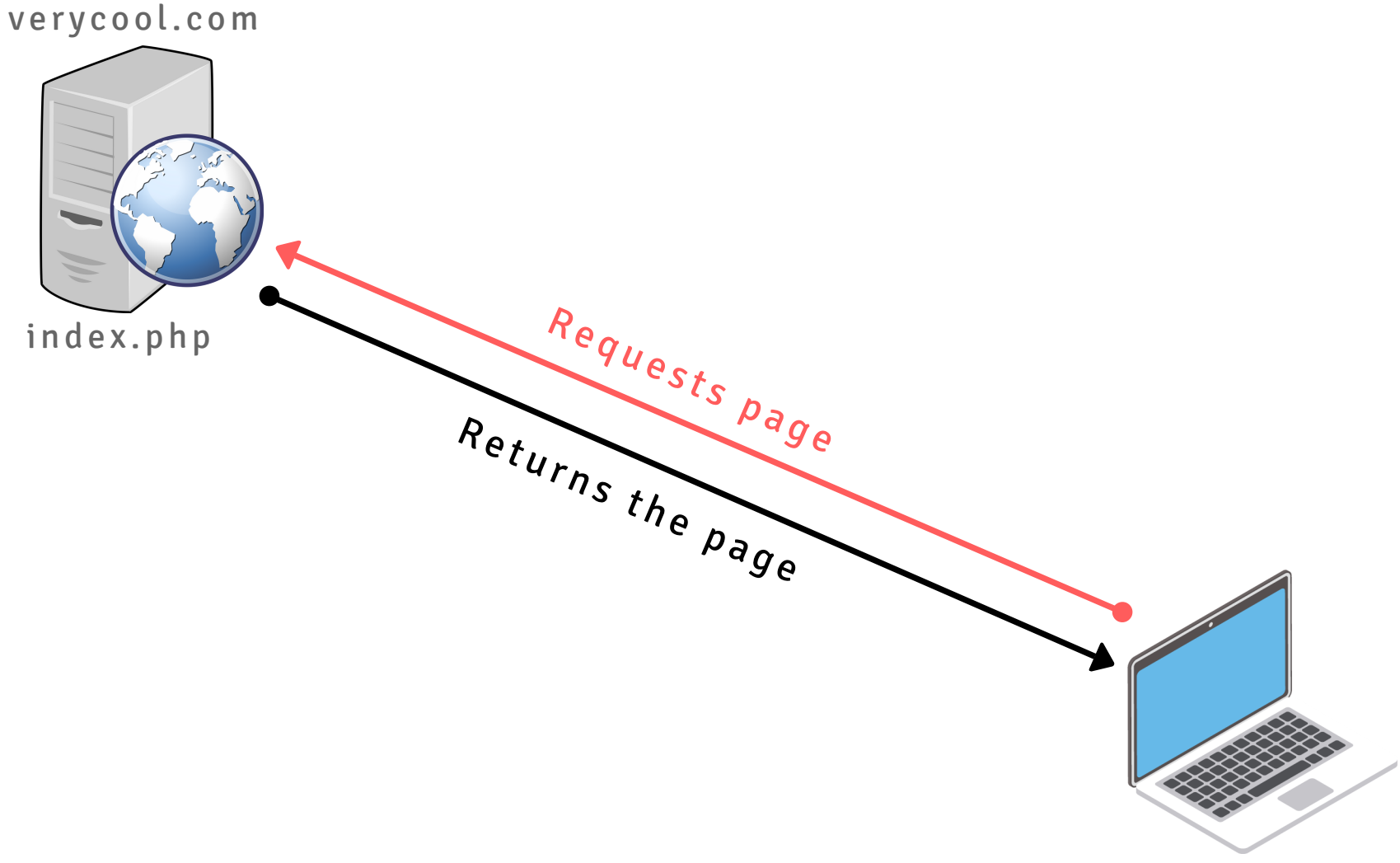
- Client makes request
- Server responds
- Clients displays / uses the response

SERVER-SIDE RENDERING

index.php

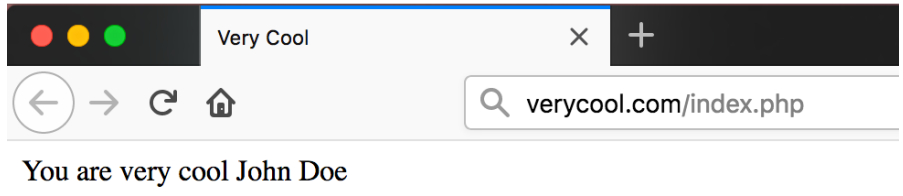
```
<html>
    <head><title>Very Cool</title></head>
    <body>
        <?php
            $mysqli = new mysqli("localhost", "my_user", "my_password", "world");

            $result = $mysqli->query("SELECT name FROM users WHERE username='johnyd'");

            echo 'You are very cool' . $result['name'];
        ?>
    </body>
</html>
```

SERVER-SIDE RENDERING



SERVER-SIDE RENDERING

verycool.com



SERVER-SIDE RENDERING

SERVER-SIDE RENDERING

Conclusion

- Server compiles the code on request
- Server returns the compiled page to the client
- Server has all the work

CLIENT-SIDE RENDERING

index.html

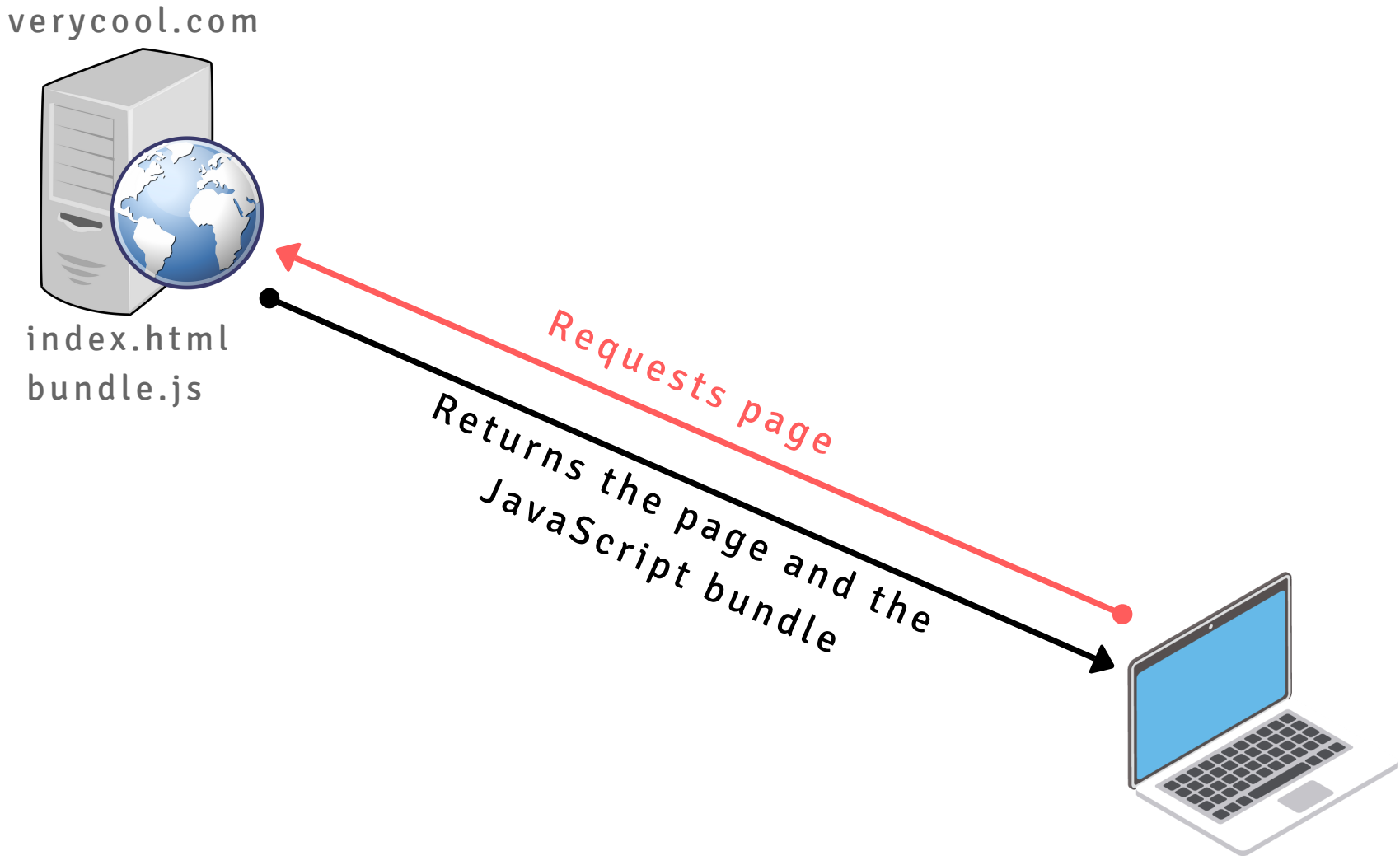
```
<html>
    <head><title>Very Cool</title></head>
    <body>
        <div id="root"></div>
        <script src="bundle.js"></script>
    </body>
</html>
```

CLIENT-SIDE RENDERING

bundle.js

```
fetch("http://api.com")
    .then(response => response.json())
    .then(name => document.getElementById("root").innerText = "You are very cool " + name);
```

CLIENT-SIDE RENDERING

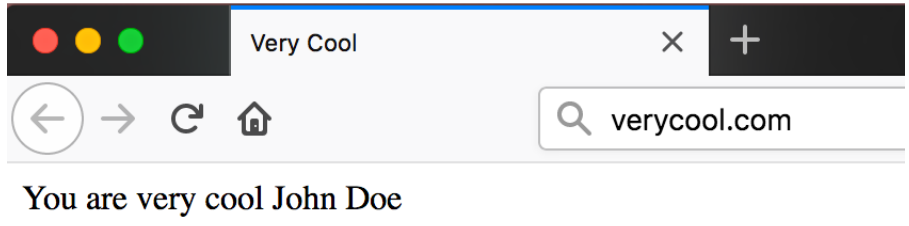


CLIENT-SIDE RENDERING

verycool.com

index.html

bundle.js



CLIENT-SIDE RENDERING

Conclusion

- Server only returns the HTML and JavaScript files
- Client renders the content through JavaScript
- Client has (**almost**) all the work
- Client needs a way to get dynamic data

APIs

Enter APIs

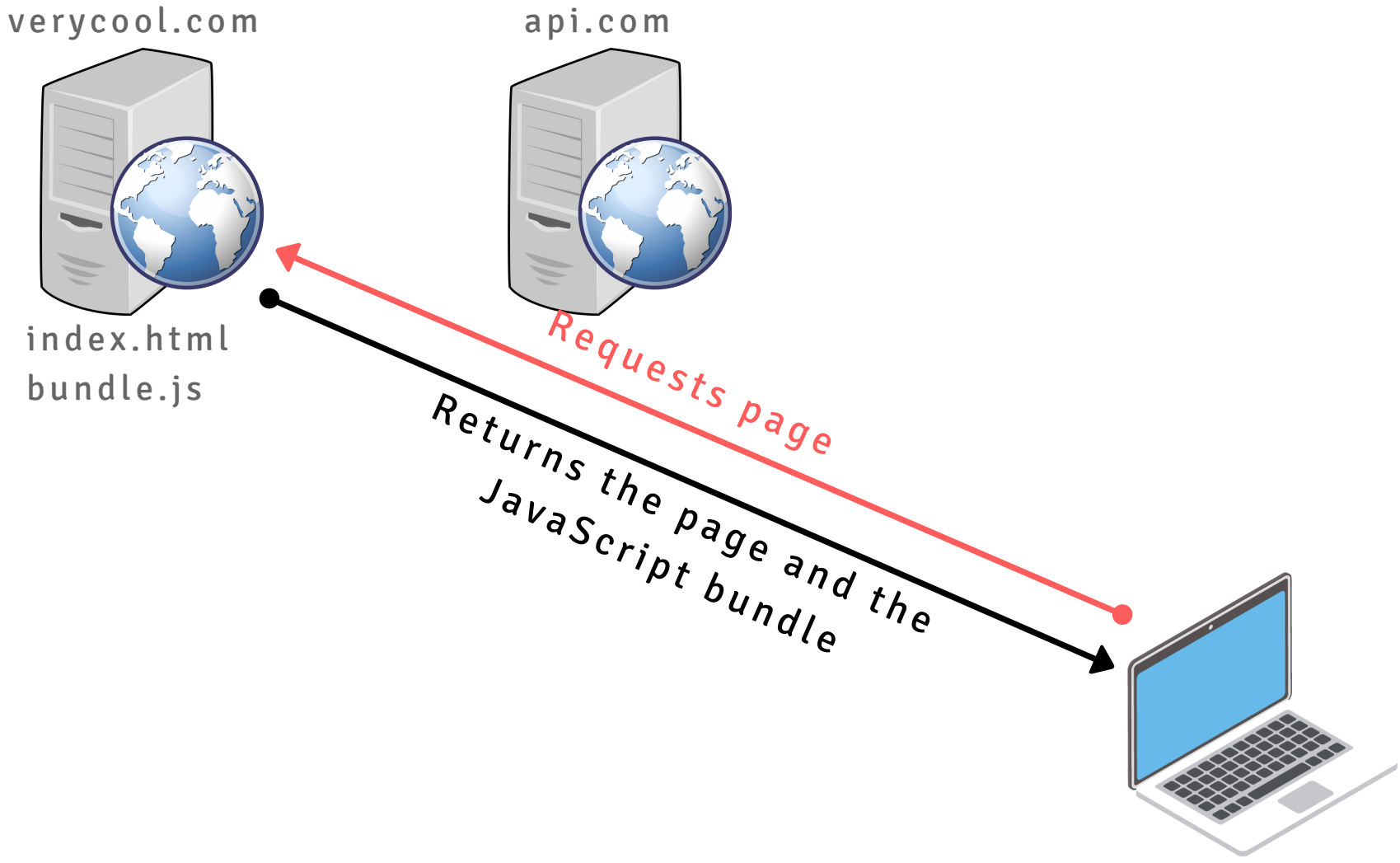
verycool.com

index.html
bundle.js

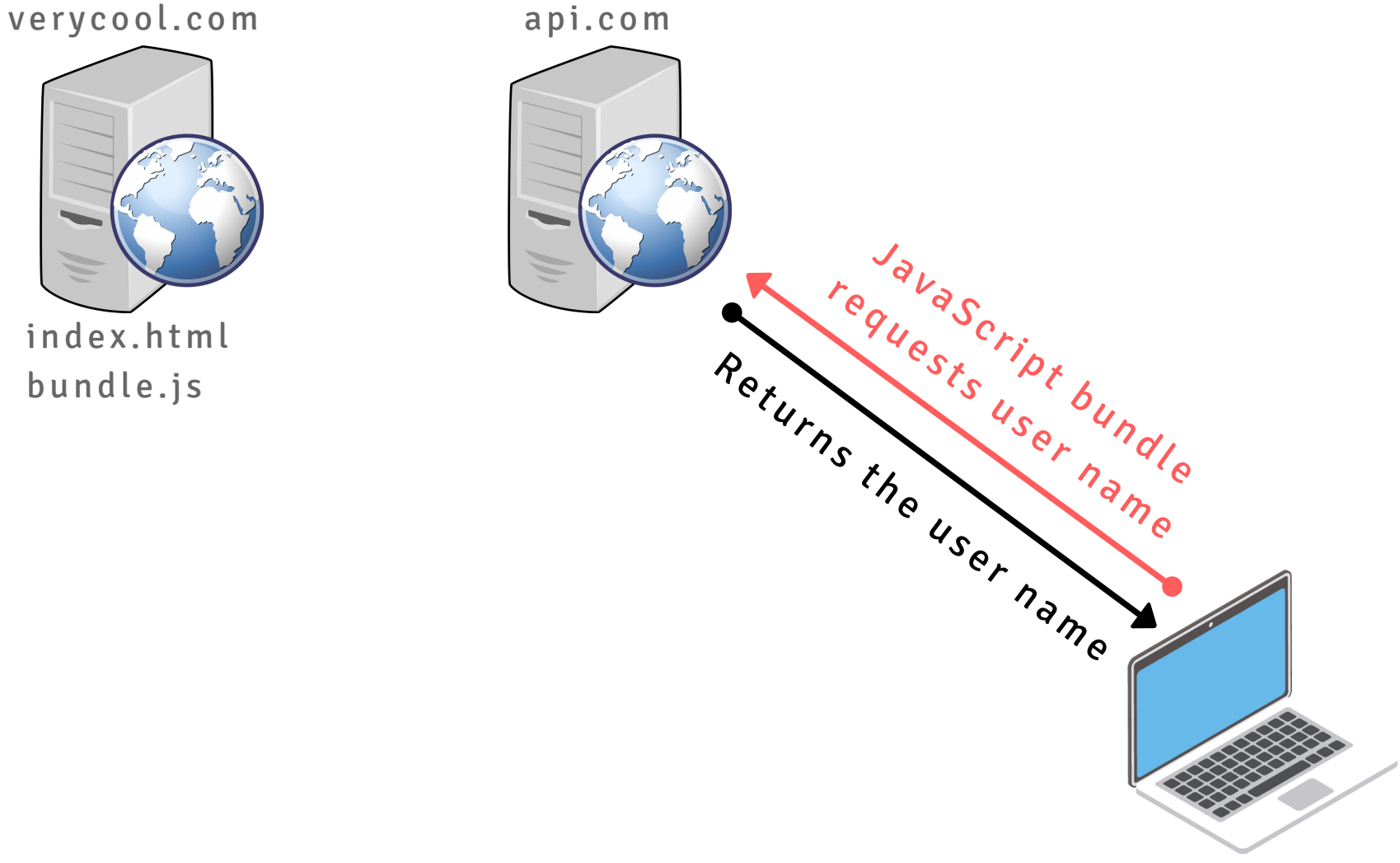
api.com

The API could be in the same server as the web app

APIs


verycool.com
api.com
index.html
bundle.js
Requests page
Returns the page and the JavaScript bundle

APIs


verycool.com
index.html
bundle.js
api.com
JavaScript bundle
requests user name
Returns the user name

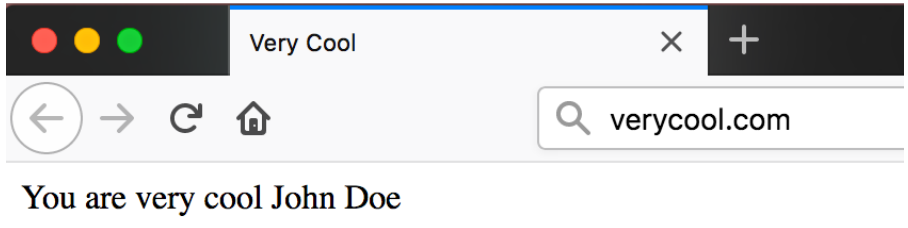
APIs

verycool.com

index.html
bundle.js

api.com

JavaScript bundle then renders the user name



APIs

Conclusion

- APIs are programs running in servers which return content in the form of text, JSON, etc
- Can be called from any device
- Highly reusable
- Perfect for client-side rendering

REACT

Browser APIs are complex and this made client-side rendering a bit too difficult to handle, for this reason a couple of JavaScript libraries started to appear and solve some of these problems.

A good and common used example is jQuery

jQuery is a fast, small, and feature-rich JavaScript library (...) with an easy-to-use API that works across a multitude of browsers.

Source: jQuery.com

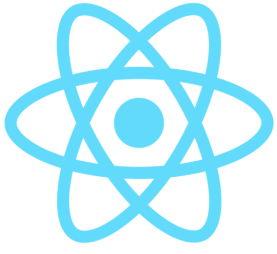
REACT

Where does React fit in all this?

React is a JavaScript library for building user interfaces. It is maintained by Facebook, Instagram and a community of individual developers and corporations.

React allows developers to create large web-applications that use data and can change over time without reloading the page. It aims primarily to provide speed, simplicity, and scalability.

Source: wikipedia



REACT

React makes use of JSX

*JSX is a syntax extension to JavaScript to describe what the UI should look like, it might sort of resemble a template language, but it comes with the full power of JavaScript. It is **faster**, **safer** and **easier**!*

```
const element = <h1>Hello, world!</h1>;
```

React embraces the fact that rendering logic is inherently coupled with other UI logic (event handling, state changes, data preparation). Instead of artificially separating concerns in different files, React separates concerns with loosely coupled units called “components” that contain both rendering logic as UI logic.

Source: reactjs

Conclusion

- JavaScript library
- Renders in client-side
- Fast, simple and scalable
- Makes use of JSX

LIVE CODING - WEB CHAT

Start by opening a terminal and typing:

```
$ npm install -g create-react-app
```

I recommend you to follow the live coding session, but all the code created during it is available at:

https://github.com/GJordao/react-workshop-ipbeja

Simply clone or download the repository, open a terminal window inside its location and type:

```
$ npm install
$ npm start
```