

Web development - Intermediate

Agenda

Day 1(How the web works):

- How the web works
- Client-server architecture
- Evolution of web(WWW vs Internet)
- Set up Developer Environment

Day 2(Bootstrap, JQuery and DOM manipulation):

- Bootstrap
- DOM
- DOM selectors and events
- jQuery

Day 3(HTTP/JSON/AJAX +Async JS):

- HTTP/HTTPS
- JSON
- AJAX
- Asynchronous JavaScript

Agenda

Day 4-5(Frameworks, React):

- Introduction about frameworks and How the frameworks work under the hood?
- Introduction to React(state, props, component)

Day 6(APIs and microservices):

- How APIs work?
- Evolution of APIs
- Micro services and Web services

Day 7(Backend):

- Basics
- Introduction to NodeJs and ExpressJs

Day 7

- Basics
- Introduction to NodeJs and ExpressJs

React JS, Angular

Node.js, PHP, Java

MySQL, MongoDB



Client



Server



Database

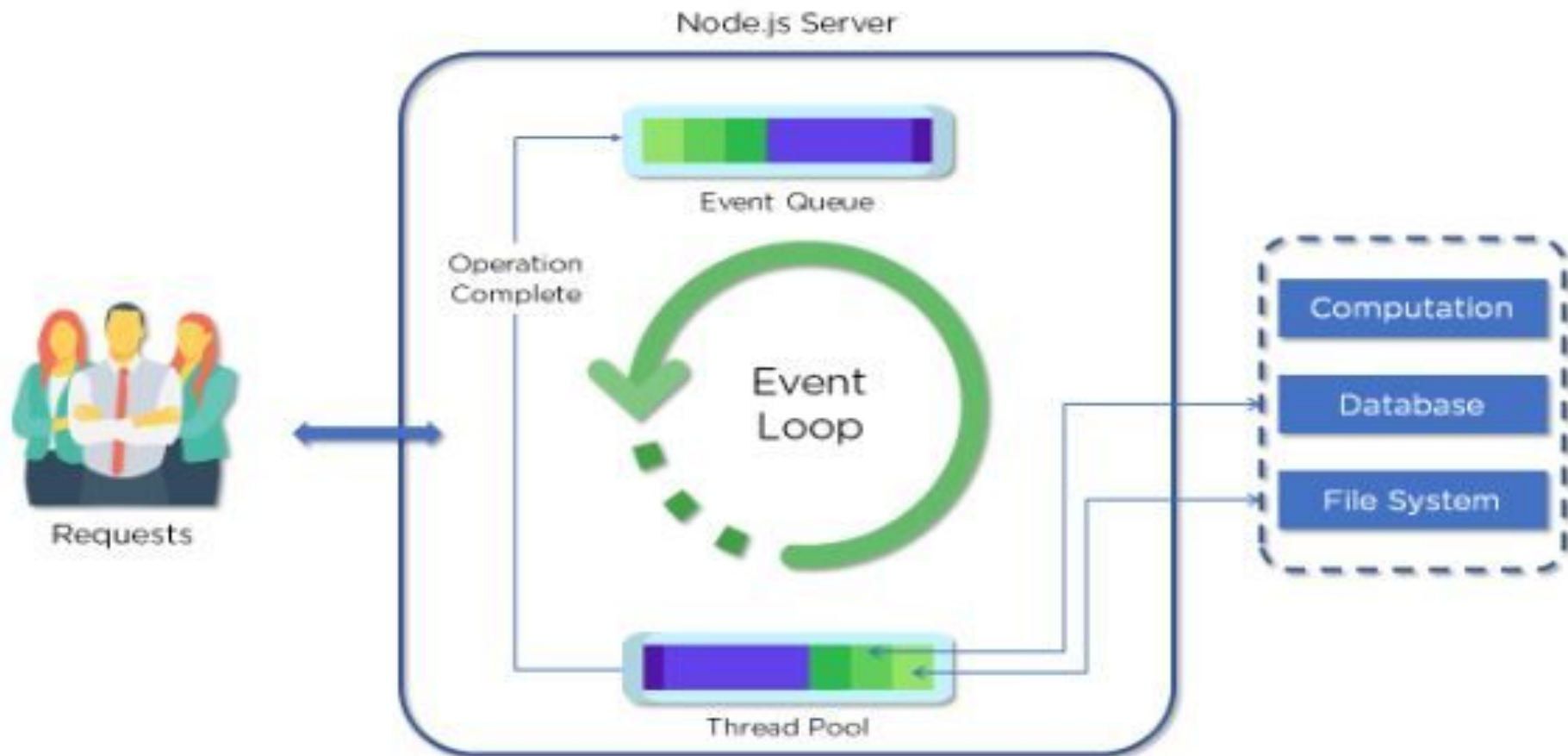
NodeJs

- open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript
- intended for use outside of a browser context
- omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries

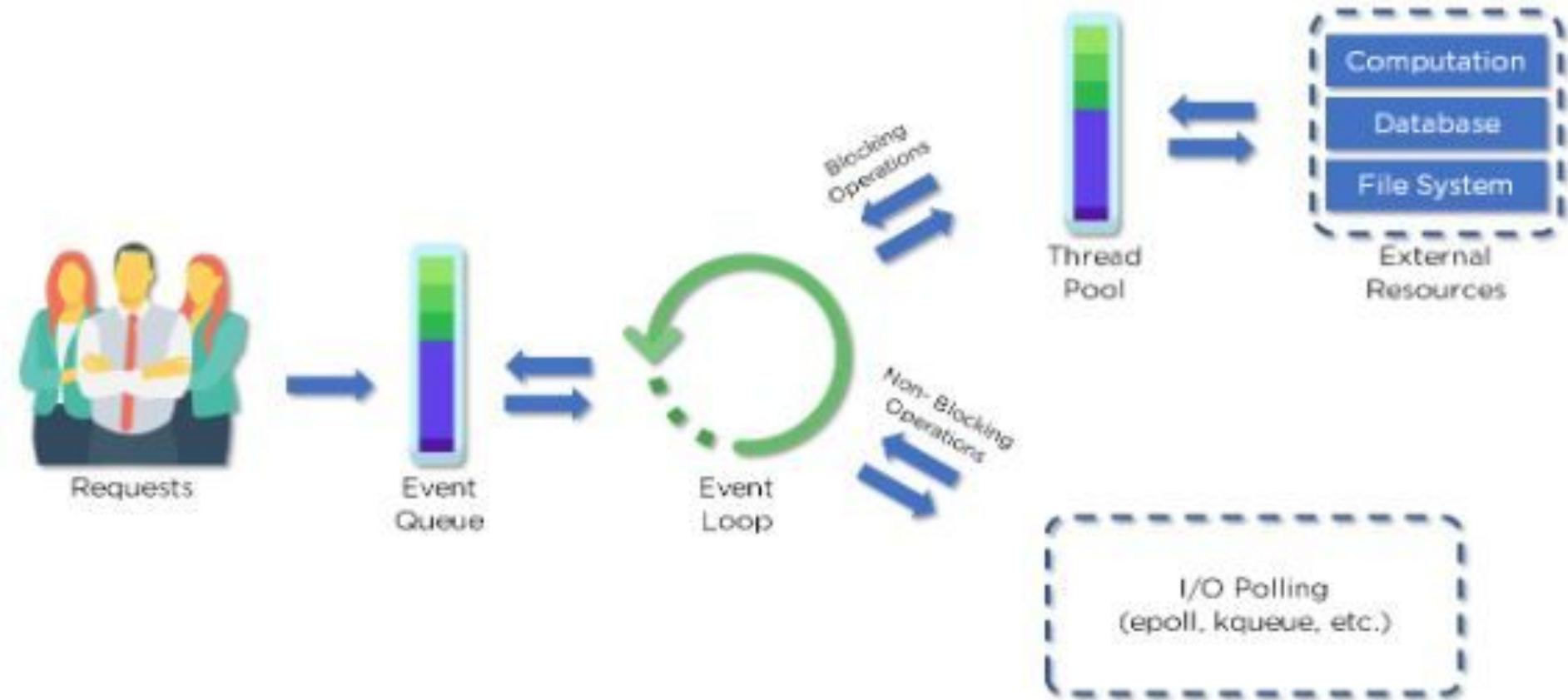
Resources :

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

NodeJs architecture



NodeJs architecture workflow



Resources for NodeJs architecture

<https://bit.ly/3aVvBy9>

Advantages of NodeJs

- Handling multiple concurrent client requests is fast and easy
- No need for creating multiple threads
- Requires fewer resources and memory

ExpressJs

- Write handlers for requests with different HTTP verbs at different URL paths (routes).
- Integrate with "view" rendering engines in order to generate responses by inserting data into templates.
- Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
- Add additional request processing "middleware" at any point within the request handling pipeline

```
var express=require('express');
```

1

use the express
module

2

```
var app=express();
```

create an object of
the express module

```
app.get('/', function (req, res) {
```

3

create a callback
function

```
res.send('Hello World!');
```

```
});
```

```
var server = app.listen(3000, function () {
```

5

Make the server
listen on port 3000

```
});
```

Send 'Hello World'
response

Further reading/Resources

<https://codeforgeek.com/express-nodejs-tutorial/>

<https://www.guru99.com/node-js-express.html>

<https://www.robinwieruch.de/getting-started-github-graphql-api>

Thank you!