Botlhale Village
Working together for ICT innovation and growth in Africa

BELGIUM CAMPUS iTversity
It's the way we're wired

WPR252

BELGIUM CAMPUS
iTversity
It's the way we're wired

Introduction to Node.js

Definition of node js

BELGIUM CAMPUS iTversity
It's the way we're wired

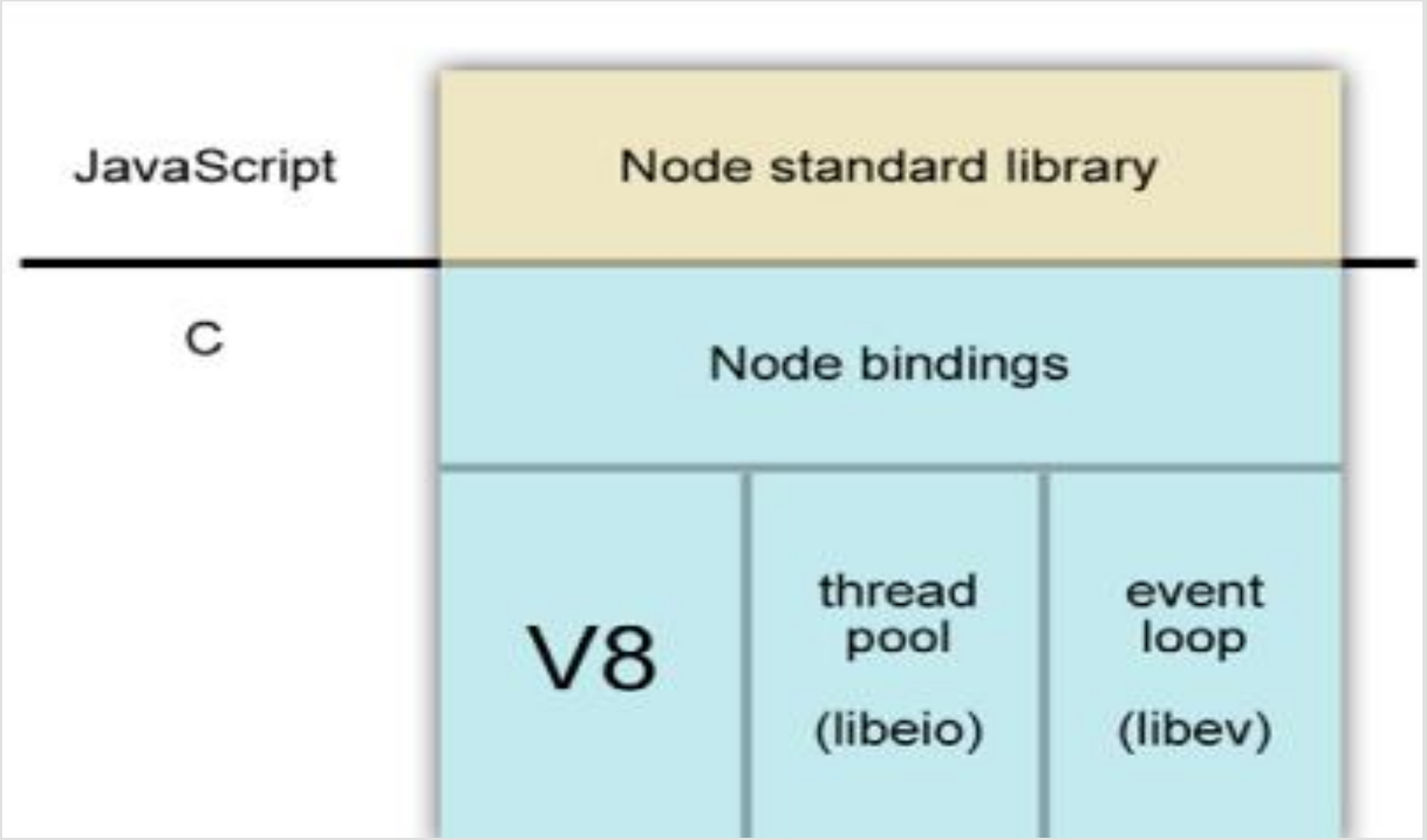
- The official website (http://www.nodejs.org) defines Node as "a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications.
- Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices."
- Node uses V8, the virtual machine that powers Google Chrome, for server-side programming.
- V8 gives Node a huge boost in performance because it cuts out the middleman, preferring straight compilation into native machine code over executing bytecode or using an interpreter.

Definition of node js

BELGIUM CAMPUS iTversity
It's the way we're wired

- Node uses JavaScript on the server, there are also other benefits:
- ❖ Developers can write web applications in one language, which helps by reducing the context switch between client and server development.
- ❖ JSON is a very popular data interchange format today and is native to JavaScript. JavaScript is the language used in various NoSQL databases (such as MongoDB), so interfacing with them is a natural fit.
- ❖ Node uses one virtual machine (V8) that keeps up with the ES5 In other words, you don't have to wait for all the browsers to catch up to use new JavaScript language features in Node.

Node.js architecture



Node.js architecture

BELGIUM CAMPUS iTversity
It's the way we're wired

In general, Node.js is made up of three things:

a) V8 is Google's JavaScript engine that is used in the Chrome web browser
b) A thread pool is the part that handles the file input/output operations. All the blocking system calls are executed here
c) The event loop library

On top of these three blocks, we have several bindings that expose low-level interfaces.

Installing node.js

BELGIUM CAMPUS iTversity
It's the way we're wired

- A fast and easy way to install Node.js is by visiting https://nodejs.org/download/ and downloading the appropriate installer for your operating system.
- For OS X and Windows users, the installer provides a nice, easy-to-use interface.
- Once complete, the installer will inform you of the two binaries it installed (node and npm)
- To make it easy to consume third-party JavaScript libraries in your applications, Node.js comes with its own package management system called *Node Package Manager* (NPM).

Packages

BELGIUM CAMPUS iTversity
It's the way we're wired

- Node.js is modular system where each file is treated as a separate module (or package).
- Every module should live in its own directory, which also contains a metadata file called package.json.
- NPM is a command-line tool that downloads and uploads Node.js packages.
- The official site, https://npmjs.org/, acts as a central registry.
- When we create a package via the npm command, we store it there so that every other developer may use it.

package.json

BELGIUM CAMPUS iTversity
It's the way we're wired

- An integral part of the NPM ecosystem is a simple JSON file called package.json.
- It is vital to have it set up properly when you want to share your module with the world, but it is just as useful if you are consuming modules from other people.
- a package.json file describes a module and It must exist if you would like to publish your package via npm.
- You can create a package file by hand, or use the handy **npm init** command-line tool, which will prompt you for a series of values and automatically generate a package.json file.

package.json values

- **name**: (Required) Holds the name of the project provided by the user
- **version**: (Required) Depicts the version of the application and must follow the semantic versioning rules.
- **description**: explanation & purpose of the application
- **entry point**: This is the file that should set module.exports — it defines where the module object definition resides.
- ***Scripts:*** Contains the list of scripts which are required to be included in the application to execute properly
- **Author***:* the name of the primary developer of the project
- **License***:* License to which the application confirms are mentioned in this key-value pair

package.json values

- ***Dependencies:*** Depicts the list of 3rd Party packages or modules installed using NPM
- ***DevDependencies:*** Dependencies that are used only in the development part of the app are specified here
- **Repository***:* detail of type & URL of the repository where the code of the application resides
- **keywords**: A comma-separated list of keywords that will help others find your module in the registry.
- **license**: Node is an open community that likes permissive licenses. "MIT“ and "BSD" are good ones here

dependencies

- It is likely that a given module will itself depend on other modules.
- These dependencies are declared within a package.json file using four related properties,which are:

❖ **dependencies**: The core dependencies of your module should reside here.

❖ **devDependencies**: Some modules are only needed during development (and not in production), such as those used for testing.

❖ **bundledDependencies**: lock a certain bundle of dependencies into a single bundled file and have those published with your package.

❖ **optionalDependencies**: Contains modules that are optional.

dependencies

BELGIUM CAMPUS iTversity
It's the way we're wired

- npm install will always install both dependencies and devDependencies.
- Example

```
"dependencies" : {
"express" : "3.3.5“,
"foo" : "http://foo.com/foo.tar.gz",
"herder": "git://github.com/sandro/herder.git#master"
}
```

Global vs. local package installation

BELGIUM CAMPUS iTversity
It's the way we're wired

- There are two main ways of installing a package using npm: locally and globally.
- *Locally* installing a package puts the downloaded module into a folder called node_modules in the current working directory and will be accessible by current your project.

  If this folder doesn't exist, npm will create it.

  Here's an example of installing the express package locally:

  npm install express

Global vs. local package installation

BELGIUM CAMPUS iTversity
It's the way we're wired

- *Globally* installing a package puts the downloaded modules at a single place in your system irrespective of the place where you execute your run command
- They are called global packages as they can be used by any of the projects present in your system.

  npm install -g express
- If you don't have sufficient file permissions when installing globally, you may have to prefix your command with sudo. sudo npm install -g express
- The major difference between local and global packages is that global are used for anything that is needed to be accessed from the shell where as local packages is typically limited to your applications or projects.

Saving Dependencies

- Whenever you run npm install, you have an optional command line flag available (--save) that tells NPM to write the information about what you installed into package.json npm install underscore –save
- If you run install with –-save, not only will it download underscore into node_modules, it will also update dependencies inside package.json to point to the installed version of underscore,
- Remove a dependency using npm rm. For example,

npm rm underscore --save deletes the underscore folder from node_modules locally and modifies the dependencies section of your package.json.

Creating simple web application

BELGIUM CAMPUS iTversity
It's the way we're wired

1. Create a new folder name it demo.
2. open the command prompt and navigate to your project directory (demo folder)
3. type in the following command. npm init
4. As soon as you hit enter, you will be asked a few questions related to your project file and the details will be stored in package.json file
5. the next step is to create the entry point of your application
6. create a JavaScript file and type in the following code and name the file as you have specified in the json file:
7. type in the following on command prompt: node nameoffile.js

Creating simple web application

BELGIUM CAMPUS iTversity
It's the way we're wired

```
const http = require('http');
const hostname = 'localhost';
const port = 8080;
const server = http.createServer((request, response) => {
    response.statusCode = 200;
    response.setHeader('Content-Type', 'text/plain');
    response.end('Welcome to Web Programming 252');
});

server.listen(port, hostname, () => {
    console.log(`Server running at http://${hostname}:${port}/`);
});
```