

JAVASCRIPT COURSE

PART 10 – 23.11.2017.



IN THIS CLASS

- Introduction to Node.js

BEFORE WE BEGIN

Presentations and homework:

<https://github.com/JSBelgrade/course-2017>

PREVIOUS HOMEWORK

PART 1:

What is a server?

What is a database?

PART 2:

What is Node.js?

What is NPM?

PART 3:

Write a Node.js Hello World.

WHAT IS A SERVER

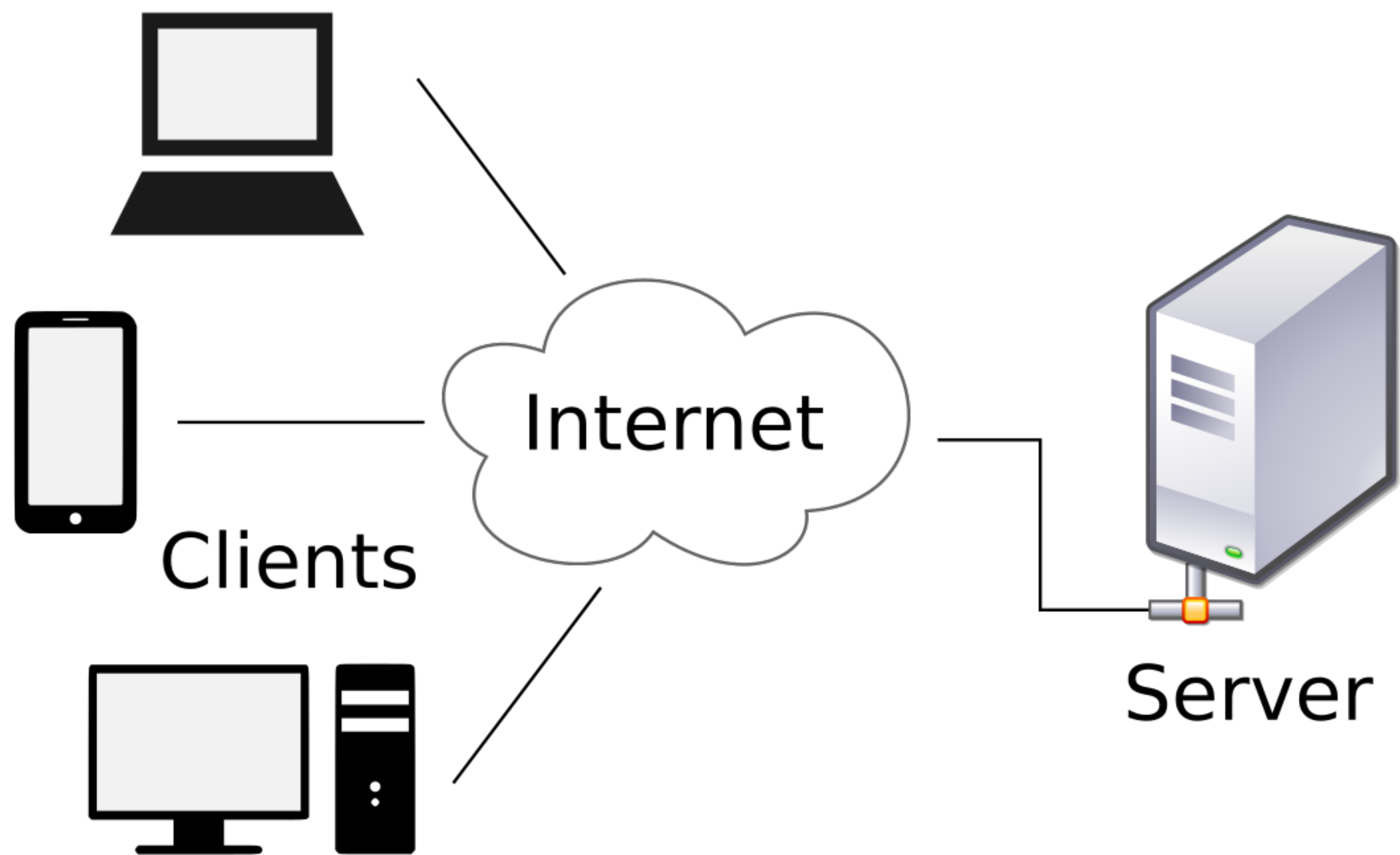
A server is a computer program that provides services to other computer programs (and their users) in the same or other computers.

WEB SERVER

A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web.

The primary function of a web server is to store, process and deliver web pages to clients.

The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP).



WHAT IS A DATABASE

A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

Some databases offer ACID
(atomicity, consistency, isolation and durability)
compliance to guarantee that data is
consistent and that transactions are complete.

COMON DATABASE TYPES

Relational vs Non-relational (NoSQL)

Relational Databases Pros

- Relational databases work with structured data.
- They support ACID transactional consistency and support “joins.”
- They come with built-in data integrity and a large eco-system.
- Relationships in this system have constraints.
- There is limitless indexing. Strong SQL.

Relational Databases Cons

- Relational Databases do not scale out horizontally very well (concurrency and data size), only vertically, (unless you use sharding).
- Data is normalized, meaning lots of joins, which affects speed.
- They have problems working with semi-structured data.

Non-Relational Databases Pros

- They scale out horizontally and work with unstructured and semi-structured data.
- Schema-free or Schema-on-read options.
- High availability.
- Many NoSQL databases are open source and “free”.

Non-Relational Databases Cons

- Weaker or eventual consistency (BASE) instead of ACID.
- Limited support for joins.
- Data is denormalized, requiring mass updates (i.e. product name change).
- Does not have built-in data integrity (must do in code).
- Limited indexing.

ACID VS BASE

Atomic – All operations in a transaction succeed or every operation is rolled back.

Consistent – On the completion of a transaction, the database is structurally sound.

Isolated – Transactions do not contend with one another. Contentious access to data is moderated by the database so that transactions appear to run sequentially.

Durable – The results of applying a transaction are permanent, even in the presence of failures.

Basic Availability – The database appears to work most of the time.

Soft-state – Stores don't have to be write-consistent, nor do different replicas have to be mutually consistent all the time.

Eventual consistency – Stores exhibit consistency at some later point (e.g., lazily at read time).

POPULAR DATABASES

SQL

- MySQL / MariaDB
- PostgreSQL
- Microsoft SQL Server

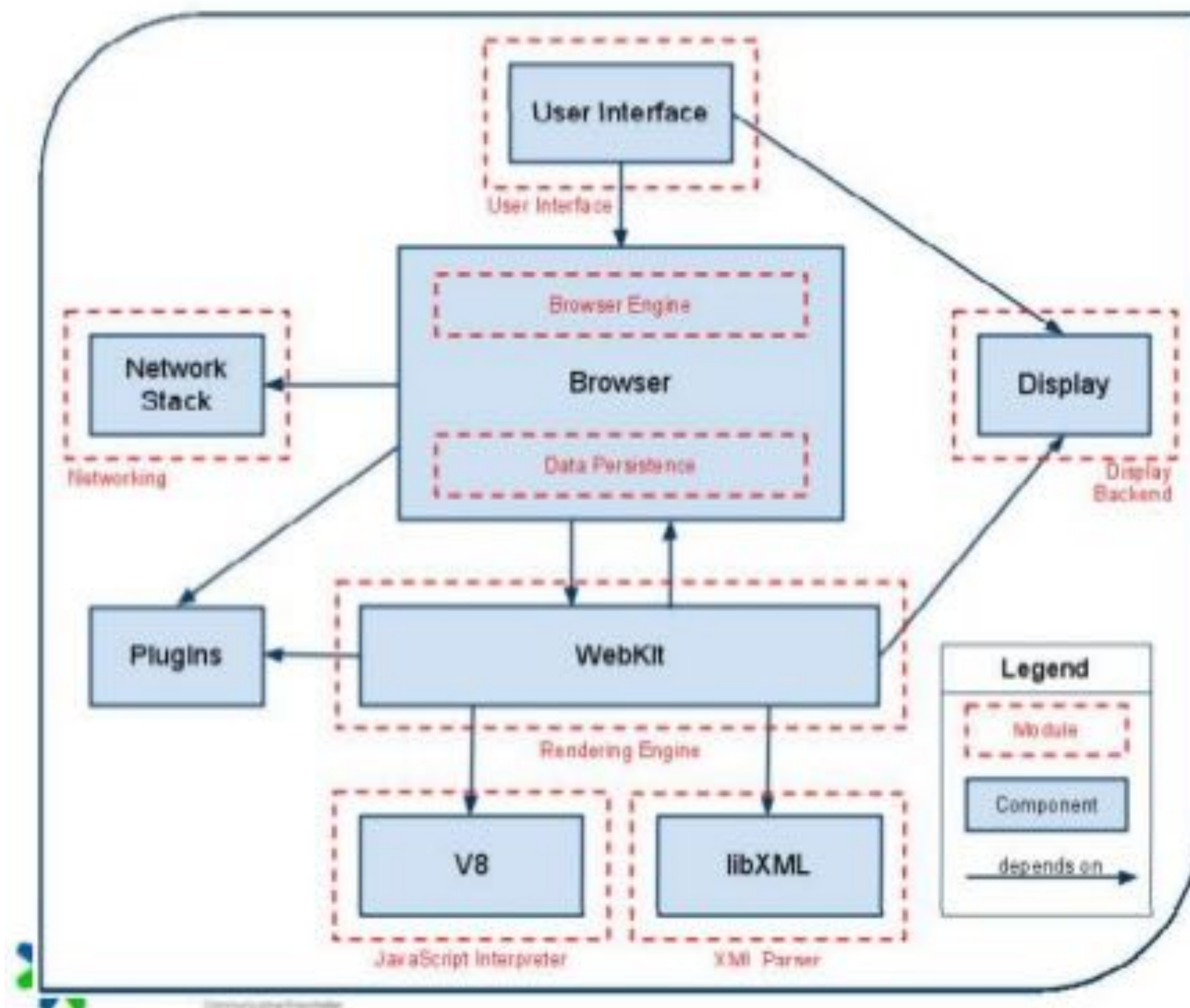
NoSQL

- MongoDB
- CouchDB
- Redis

WHAT IS NODE.JS

Node.js is a JavaScript runtime built on
Chrome's V8 JavaScript engine.

Architecture of Chrome



Rendering Engine:

Used the WebKit until v27, from v28 user WebKit fork Blink

XML Parser:

libXML to parse XML

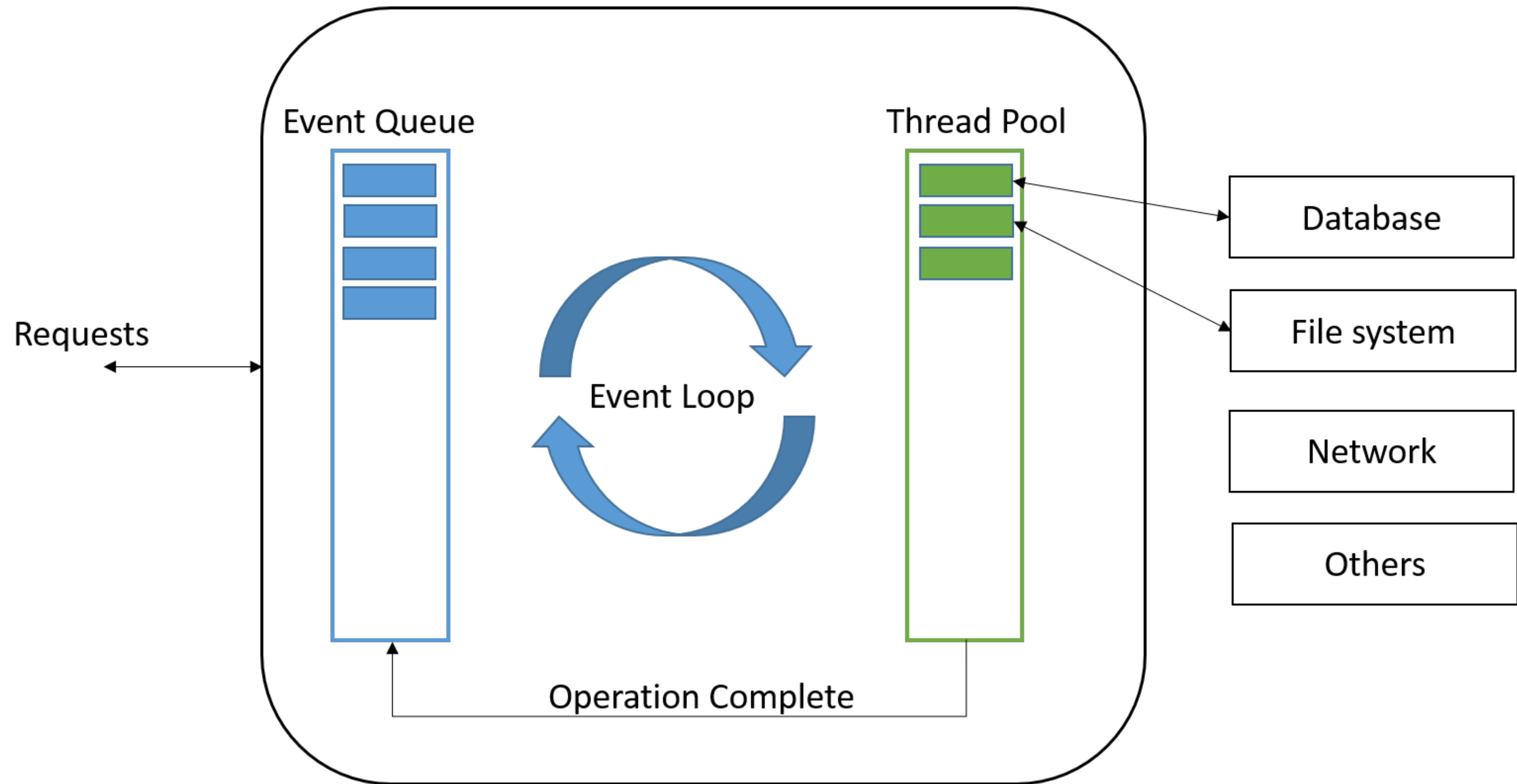
libXSLT to handle XSLT

JavaScript Interpreter: V8

JavaScript Engine, written in C++

Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

Node.js Server



WHAT IS EVENT LOOP?

"What the heck is the event loop anyway?"
by Philip Roberts at JSConf EU 2014

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

WHAT IS NPM

Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

npm makes it easy for JavaScript developers to share the code that they've created to solve particular problems, and for other developers to reuse that code in their own applications.

INSTALL NODE.JS AND NPM

Visit:

<https://nodejs.org/>

NODE.JS HELLO WORLD

hello.js

```
console.log( 'Hello world! ' );
```

Run:

```
$ node js-kurs/hello.js
```

HELLO WORLD SERVER

server.js

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200);
  res.end('Hello from server');
});

server.listen(8080);
```

Run:

```
$ node server.js
```

NodeSchool #5

December 2, 2017

Registration form:

<https://jsbelgrade.typeform.com/to/QDbZPE>

READ (AND LEARN) MORE

NodeSchool

<https://nodeschool.io>

Free Code Camp
Learn to code for free.

<https://www.freecodecamp.org>

Eloquent JavaScript

Marijn Haverbeke

<https://eloquentjavascript.net>

You Don't know JavaScript

Kyle Simpson

<https://github.com/getify/You-Dont-Know-JS>

JavaScript: The Definitive Guide

David Flanagan

<http://shop.oreilly.com/product/9780596805531.do>

JavaScript: The Good Parts

Douglas Crockford

<http://shop.oreilly.com/product/9780596517748.do>

THE END

OF PART TWO