1. **Why should clamscan be called from C++ not within NodeJS?**
2. **Justification of shifting the task to C++ server instead of doing the same on the NodeJS.**

**Few words about NodeJS from Official Documentation:**

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

**Reasons:**

1. **CPU intensive tasks may block the NodeJS server to take request from the client.**

Tasks like scanning large file is a CPU intensive tasks. As described above, NodeJS is very much efficient for I/O operations but for CPU intensive tasks, it may get busy with the execution of current task that delays the serve other web requests from the client.

**Code to demonstration**:

**const express = require('express');**

**const app = express();**

**app.get('/', (req,res)=>{**

**res.send("Hello from server");**

**});**

**app.listen(3000, ()=>{**

**console.log("server is running on port 3000");**

**});**

**for(var i=0;i<10;i++){**

**setTimeout(()=>{**

**for(var x = 0; x<100000;x++){**

**for(var y = 0;y<100000;y++){**

**}**

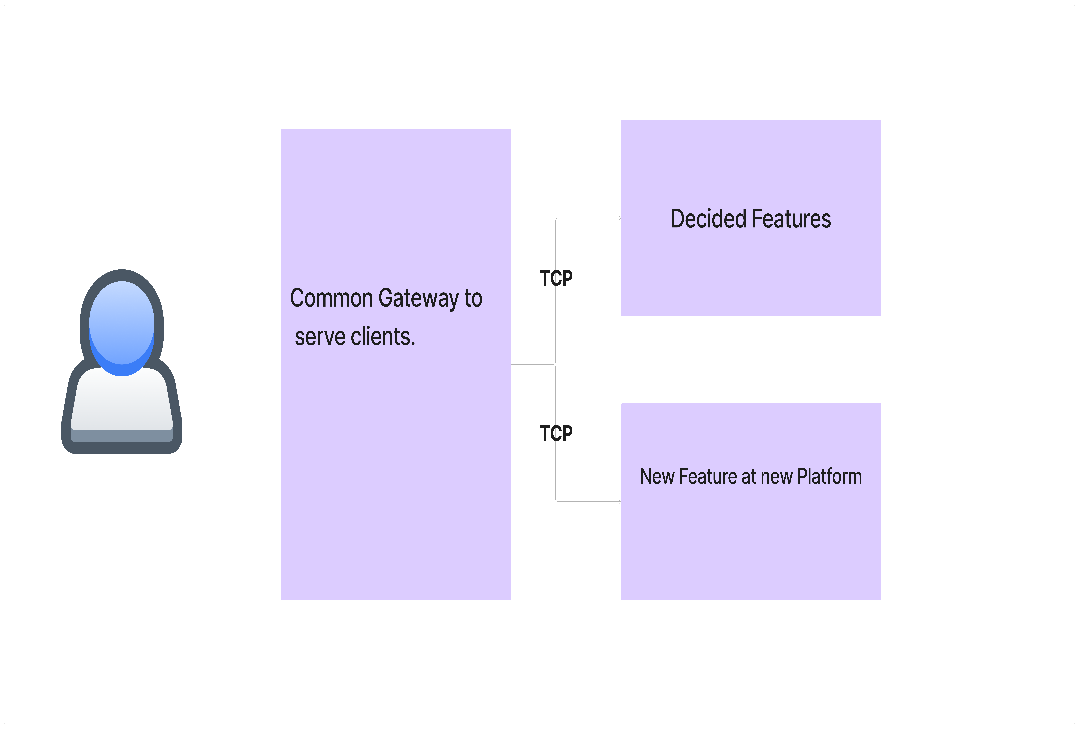
**}**

**console.log("Timeout finished");**

**}, 0);**

**}**

1. **Separating the Codebase such that adding new feature on different platform will not affect the current functions**. Adding features in future like Behavioural scanning, Network scanning etc may require different platform like Python based platform etc. Since the Gateway is connected to core feature implementation via TCP connection, Adding new feature will not burden to change the gateway’s configuration. Just establishing a new TCP connection will allow Gateway to access the new Implemented feature.



1. **Compatibility with the system.**

Programming languages like C, C++, Rust etc. provides more flexibility to interact with Operating system to write system level Programming while Programming languages like Javascript, PHP etc are designed with the Idea to serve Web request more efficiently as well as make development process smooth via abstracting the internal implementation details.

The final verdict of the question “**Is segregating of tasks in different platform a good decision?**” is heavily dependent on the rigorous testing of UTM performance.