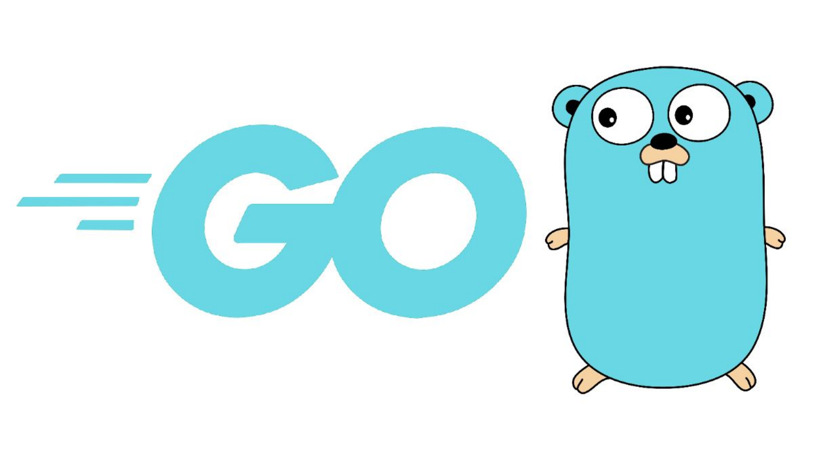
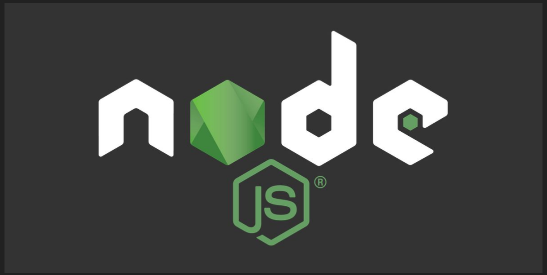
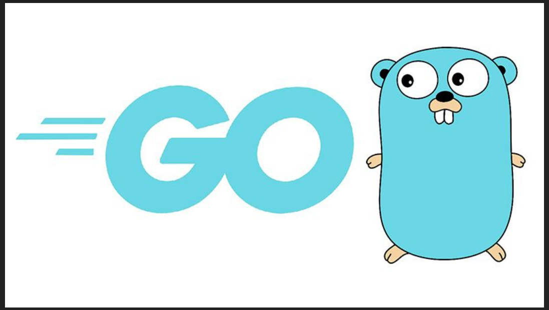
Go and backend development



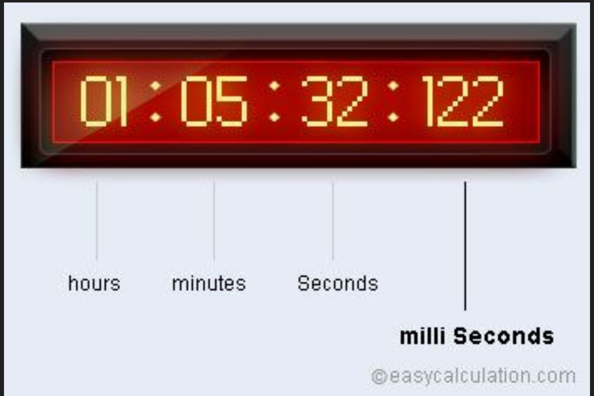
C++

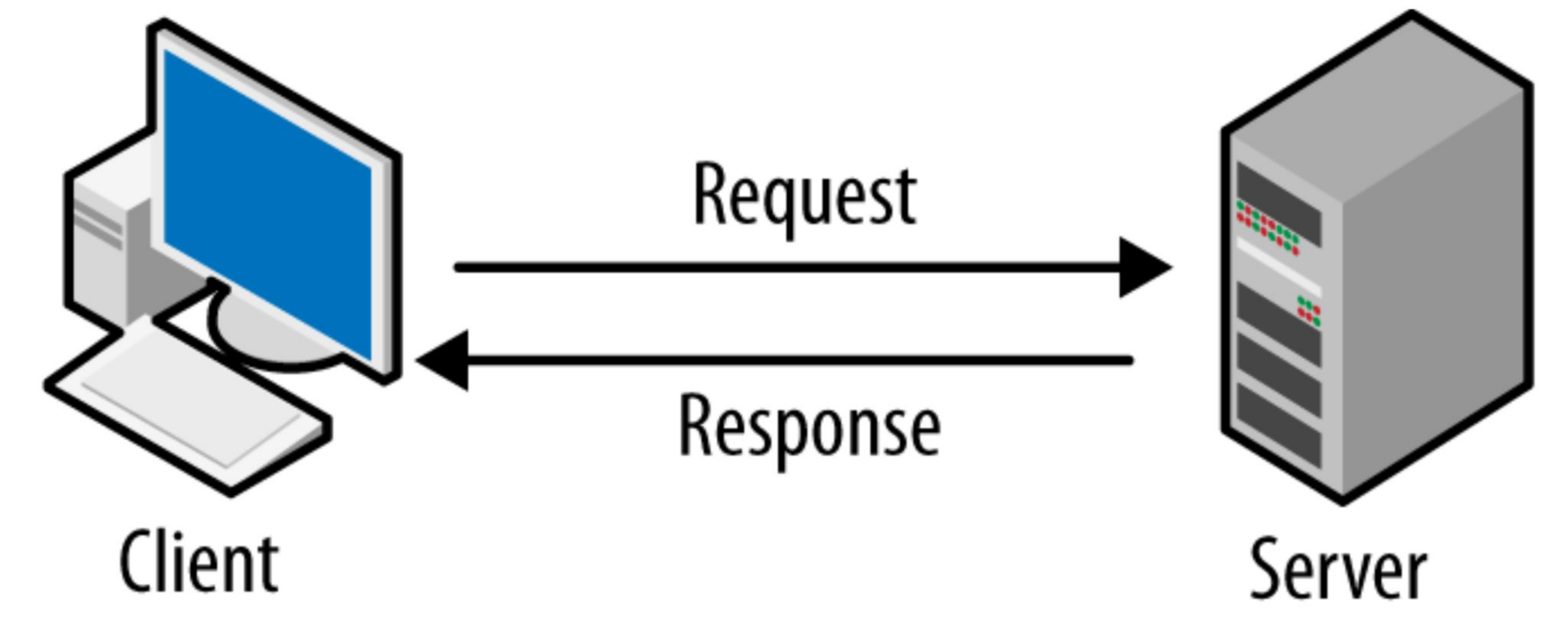

node
JS®


GO

WEB
DEVELOPMENT
We focus on Quality &
Cost-Effective solutions

EMBEDDED
SYSTEMS
PROCESSOR


01 : 05 : 32 : 122
hours
minutes
Seconds
milli Seconds
©easycalculation.com

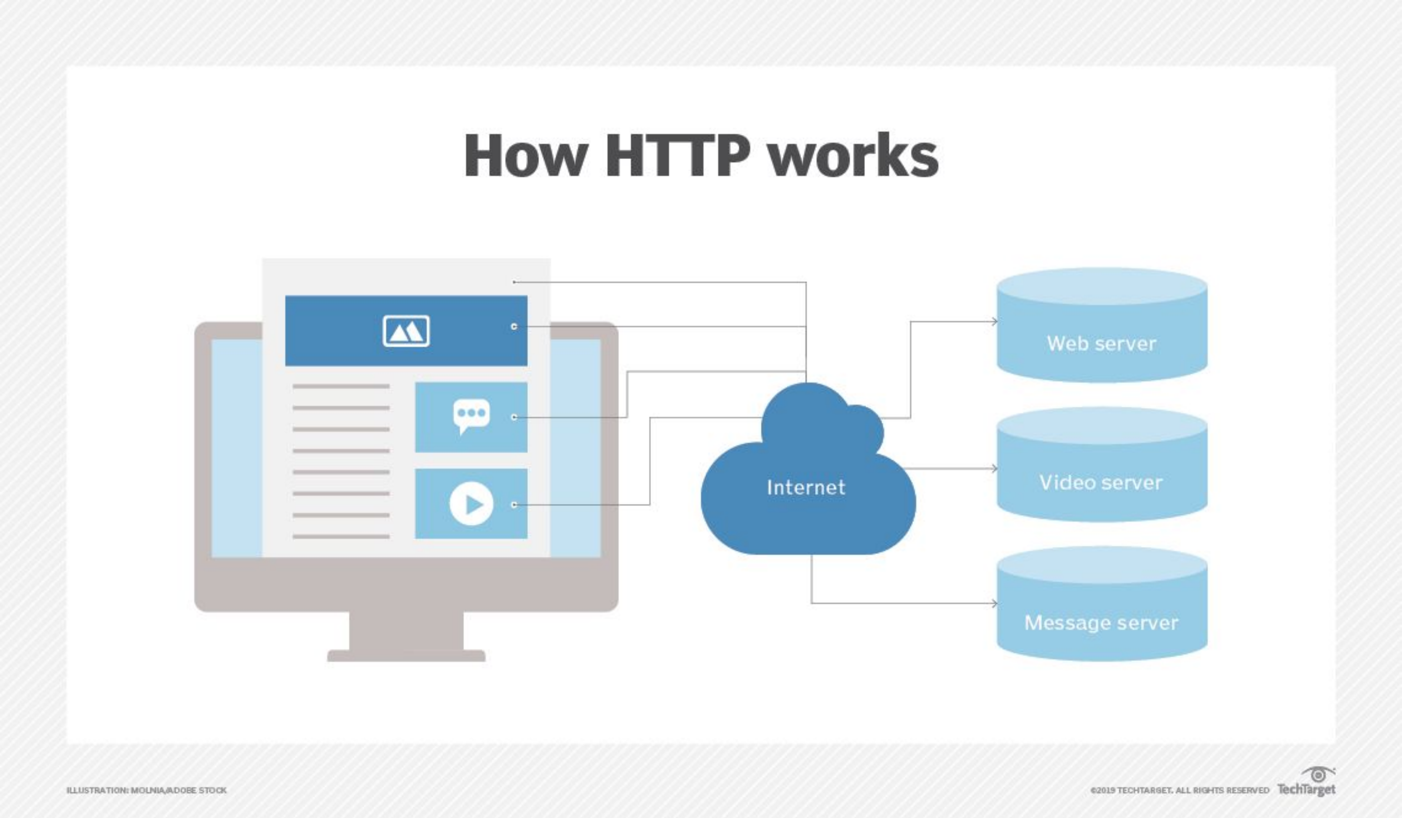

Request
Response
Client
Server

response:

Hello world 1

Hello world 2

….



How HTTP works

ILLUSTRATION: MOLNIA/ADOBE STOCK

Examples

Server

- nodejs server

Clients

- single threaded C++
- asynchronous Js
- multi threaded C++
- single threaded Go
- multi threaded Go

Summary

	C++	Js	Go
	single threaded sync	single threaded async	single threaded sync
	multi threaded sync		multi threaded sync
4000 requests get time	2.2 s	3 s	2.4 s

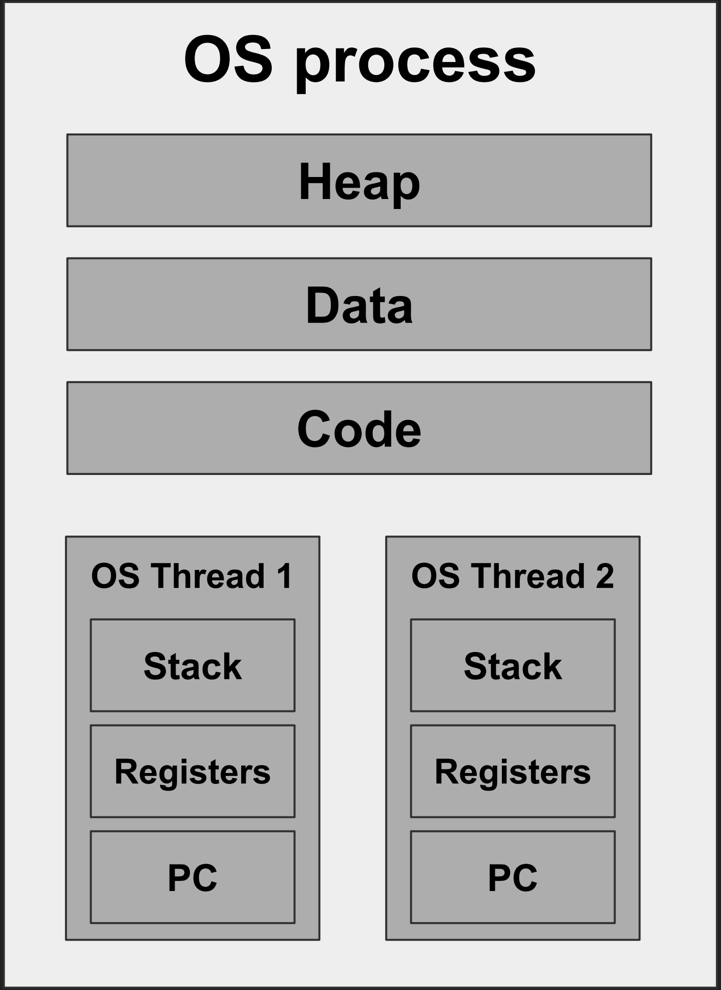
C++ vs Go multithreading

- [cpp, go]_multithreaded_calc.go

Threads

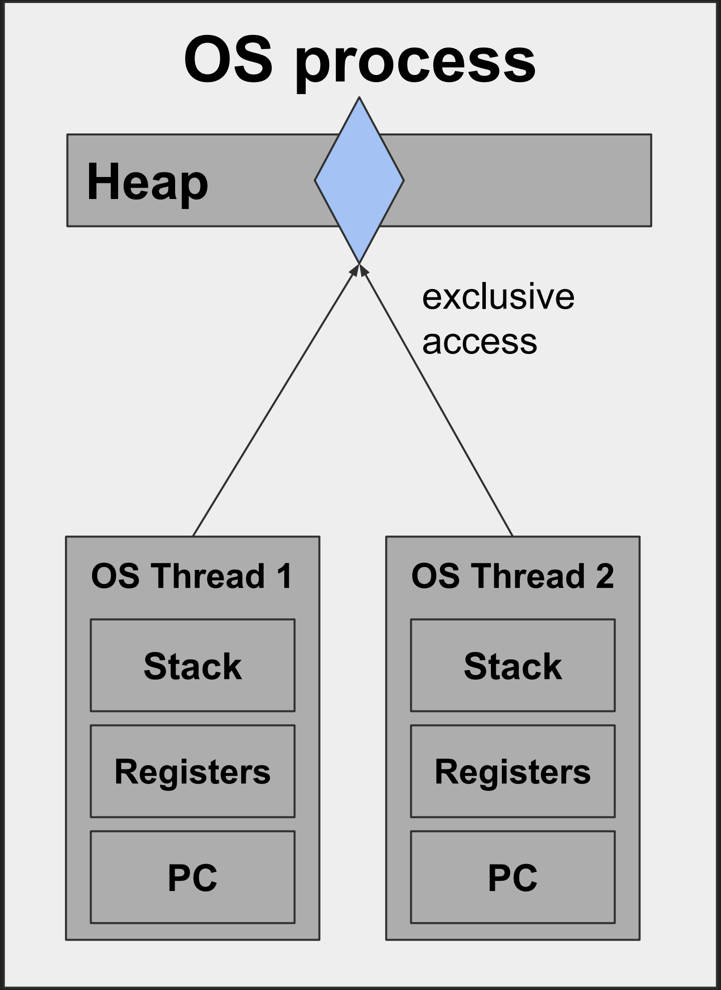
- The smallest unit of execution that CPU accepts
- There can be many threads within a process
- Each process has at least 1 thread
- Threads can run in parallel or concurrently

- Threads share the same address space
- Fixed stack size (eg. 8MB on ubuntu)
- Thread context switching is expensive - process context and thread context is copied



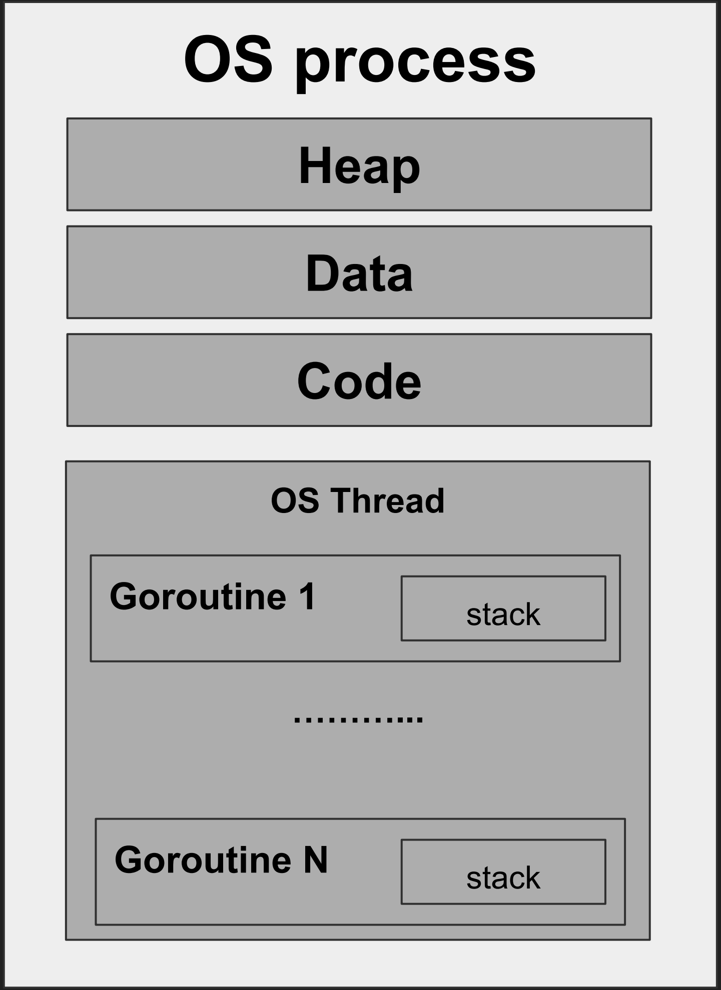
Threads

- Sharing memory between threads creates complexity and can lead to data races
- Critical section guards are a developer convention, no tool to enforce that



Goroutines

- We can think of goroutines as user space threads, managed by go runtime (built into executable)
- goroutines run in the context of a single OS thread - from OS perspective nothing has changed
- lightweight - 2KB of stack
- low CPU overhead - three instructions per function call
- sharing of memory is avoided with usage of channels



Communicating Sequential Processes

- paper by Tony Hoare (1978)
- based on 3 simple ideas:
  - each process is build for sequential execution - every process has a local state and it operates on it
  - no shared memory - data is communicated between processes - no race conditions or deadlocks
  - scale by adding more of the same

Go lang overview

- Go was developed by Google to replace the in-house use of Python, C/C++, Java and other system languages.

- First released in 2009, written in C, C alike syntax,

- Go is open-source and has a rich community that actively participates in its development.

- The language core is updated every six months while keeping the programming interface stable.

- Compiled

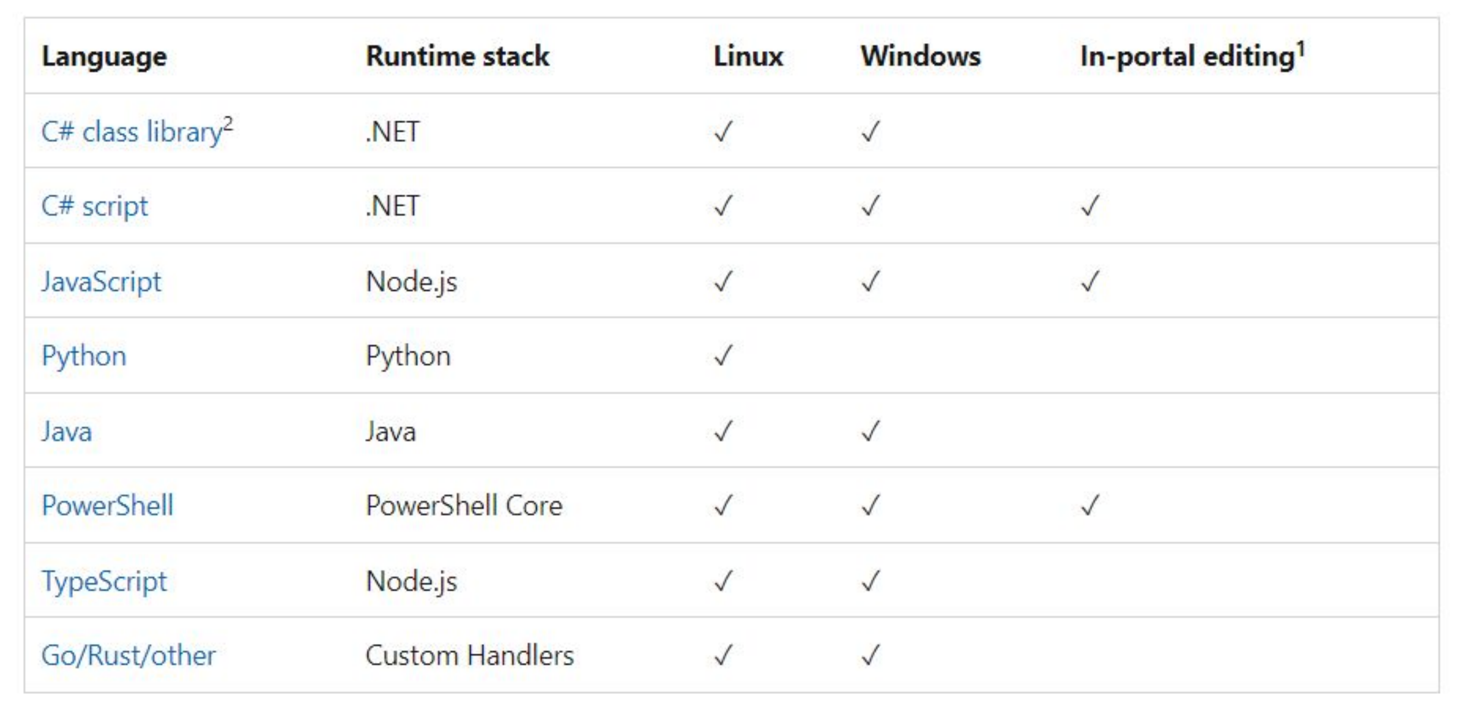
- Statically typed

- procedural/functional paradigm

Go lang overview

- go_overview.go
- go_overview_test.go
- go_multithreaded[2,3,4].go

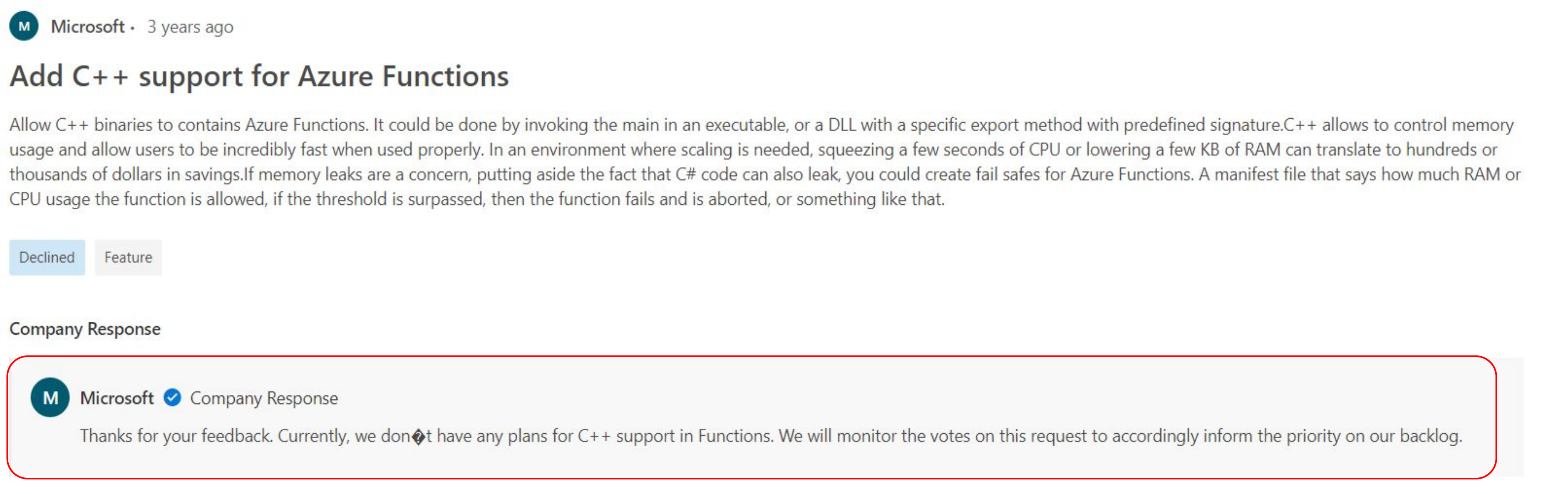
Cloud development



Language	Runtime stack	Linux	Windows	In-portal editing[1]
C# class library[2]	.NET	✓	✓	
C# script	.NET	✓	✓	✓
JavaScript	Node.js	✓	✓	✓
Python	Python	✓		
Java	Java	✓	✓	
PowerShell	PowerShell Core	✓	✓	✓
TypeScript	Node.js	✓	✓	
Go/Rust/other	Custom Handlers	✓	✓	

- azure functions serverless example
- https://docs.microsoft.com/en-us/azure/azure-functions/supported-languages

How about c++?



M Microsoft · 3 years ago

Add C++ support for Azure Functions

Allow C++ binaries to contains Azure Functions. It could be done by invoking the main in an executable, or a DLL with a specific export method with predefined signature.C++ allows to control memory usage and allow users to be incredibly fast when used properly. In an environment where scaling is needed, squeezing a few seconds of CPU or lowering a few KB of RAM can translate to hundreds or thousands of dollars in savings.If memory leaks are a concern, putting aside the fact that C# code can also leak, you could create fail safes for Azure Functions. A manifest file that says how much RAM or CPU usage the function is allowed, if the threshold is surpassed, then the function fails and is aborted, or something like that.

Company Response

M Microsoft ✓ Company Response

Thanks for your feedback. Currently, we don�t have any plans for C++ support in Functions. We will monitor the votes on this request to accordingly inform the priority on our backlog.

https://feedback.azure.com/d365community/idea/84989e2f-f224-ec11-b6e6-000d3a4f0da0

Summary

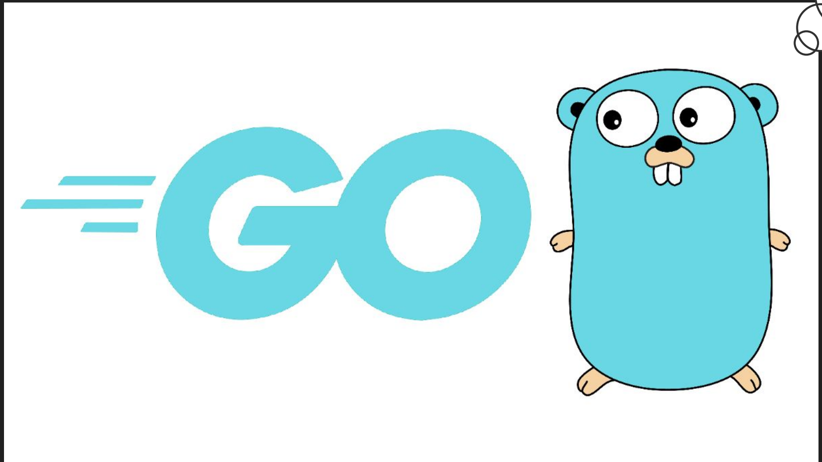
	C++	js/nodejs	go
versatile	★ ★ ★	★ ★ ★	★ ★
performance	★ ★ ★	★	★ ★
faster compile time	★	—	★ ★ ★
easy to learn/use	★	★ ★	★ ★ ★
community	★ ★ ★	★ ★ ★	★ ★

	OOP class based	OOP prototype based	post OOP
	compiled	interpreted	compiled
	statically typed	dynamically typed	statically typed

Go learning resources

- https://gobyexample.com/
- https://go.dev/doc/tutorial/
- https://motorolasolutions.udemy.com/course/concurrency-in-go-golang/

Thank you!



https://github.com/bondyr/backendDevPpt