Runtime & Concurrency in Go

V N Nikhil Anurag

GopherCon Singapore 2017

Agenda

Concurrency & Parallelism

Lifecycle of Binaries

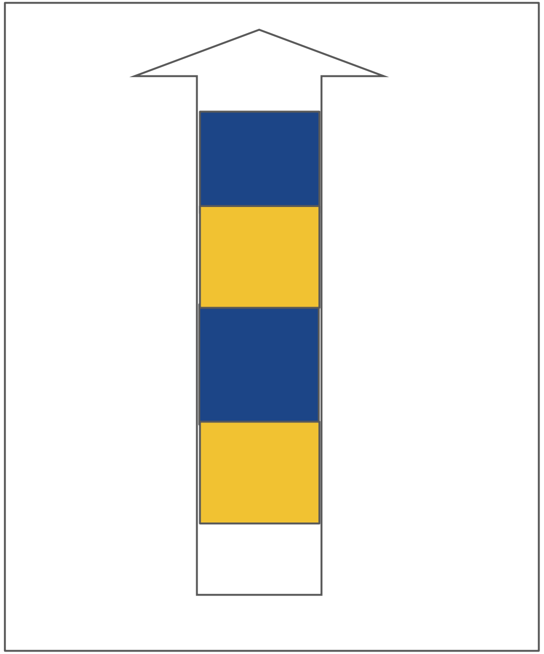
Coroutines & Green Threads

Goroutines

Pop Quiz

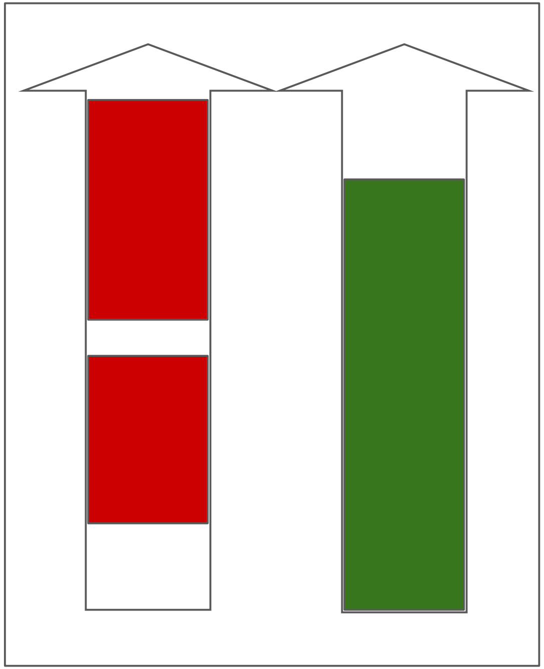
Concurrency & Parallelism

Concurrency



Concurrency is about *dealing with* lots of things at once.*

Parallelism



Parallelism is about *doing* lots of things at once.*

* https://blog.golang.org/concurrency-is-not-parallelism

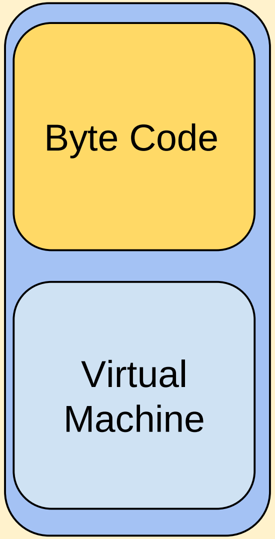
Lifecycle of Binaries

Lifcycle Of Binaries

Lifecycle of Interpreted Languages

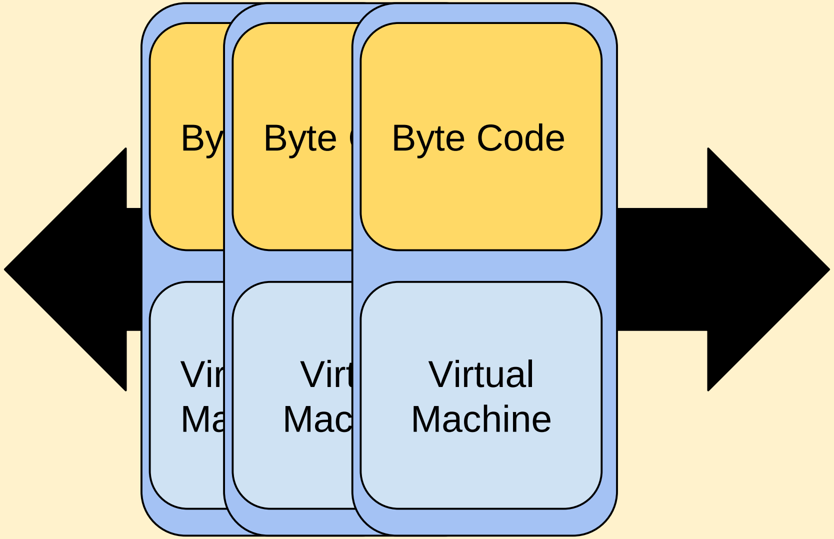
Lifecycle of a Go Binary

Lifecycle of Interpreted Languages (VM)



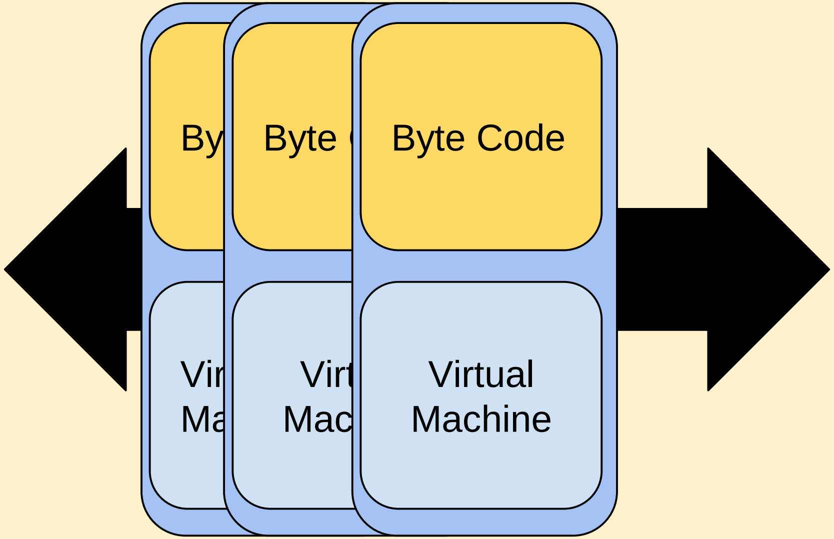
Single Executable

Lifecycle of Interpreted Languages (VM)



(VM + Application Code) * No. of Workers

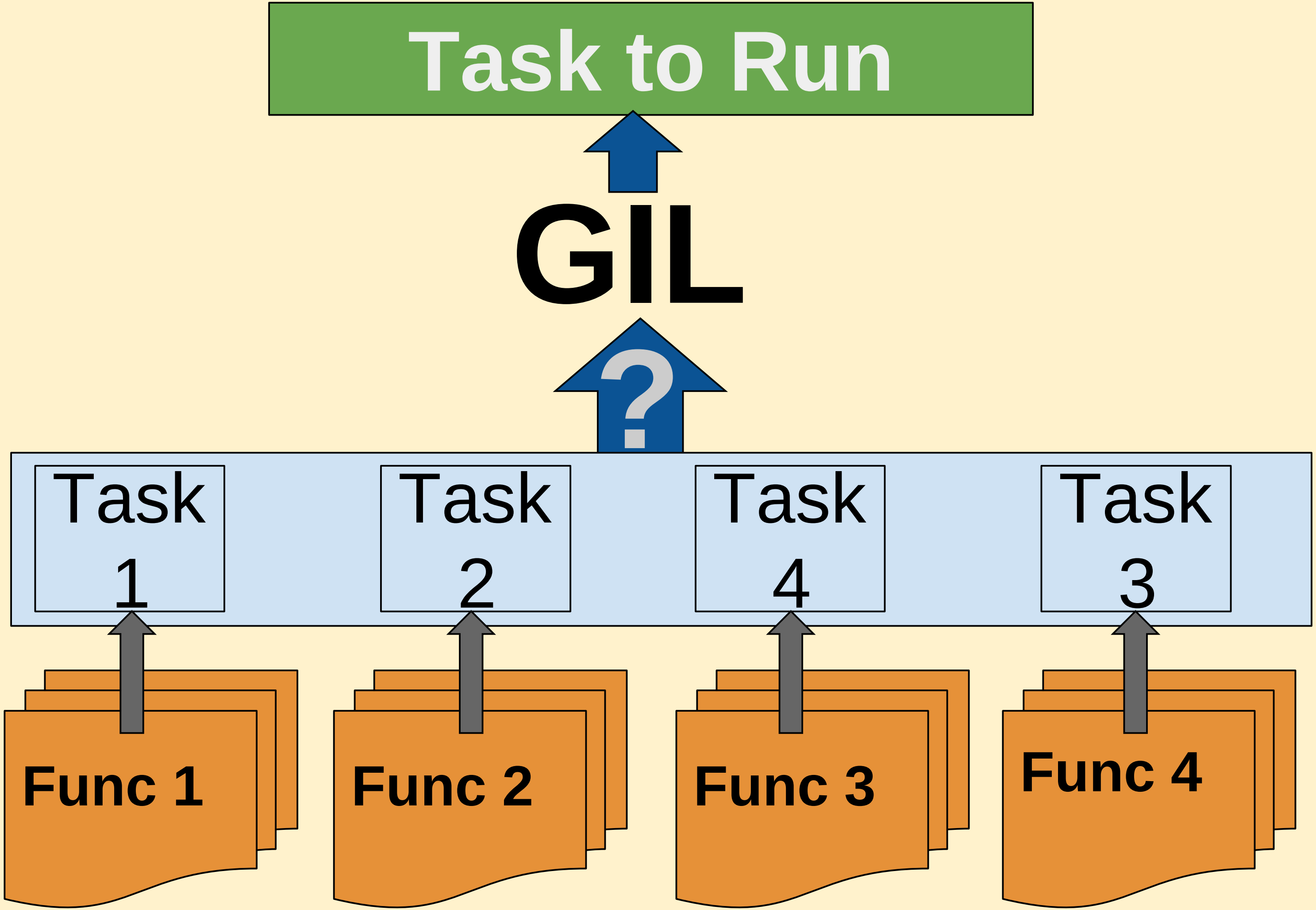
Lifecycle of Interpreted Languages (VM)



(VM + Application Code) * No. of Workers

This is required because...

Global Interpreter Lock



Task to Run

GIL

?

Task 1

Task 2

Task 4

Task 3

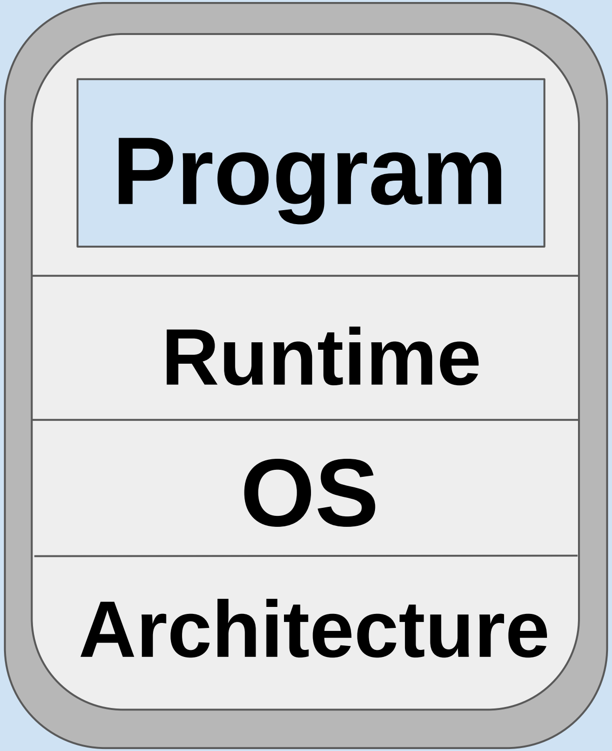
Func 1

Func 2

Func 3

Func 4

A Go Binary



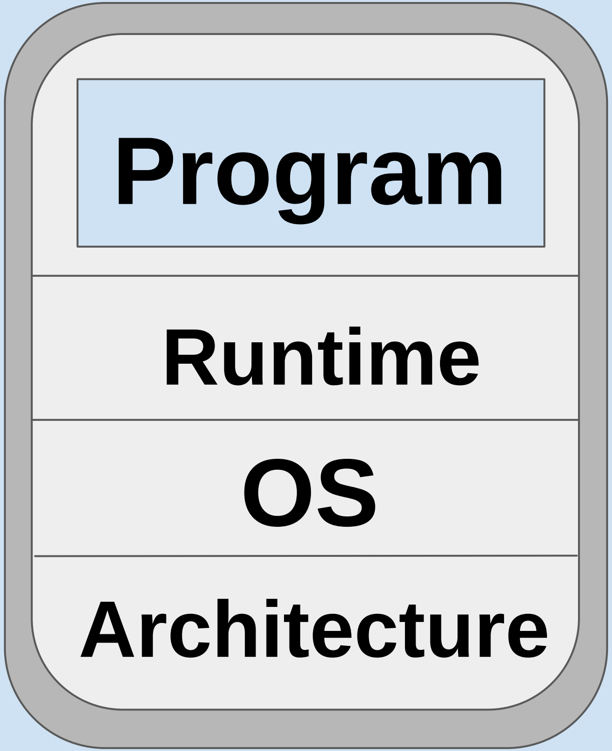
Program

Runtime

OS

Architecture

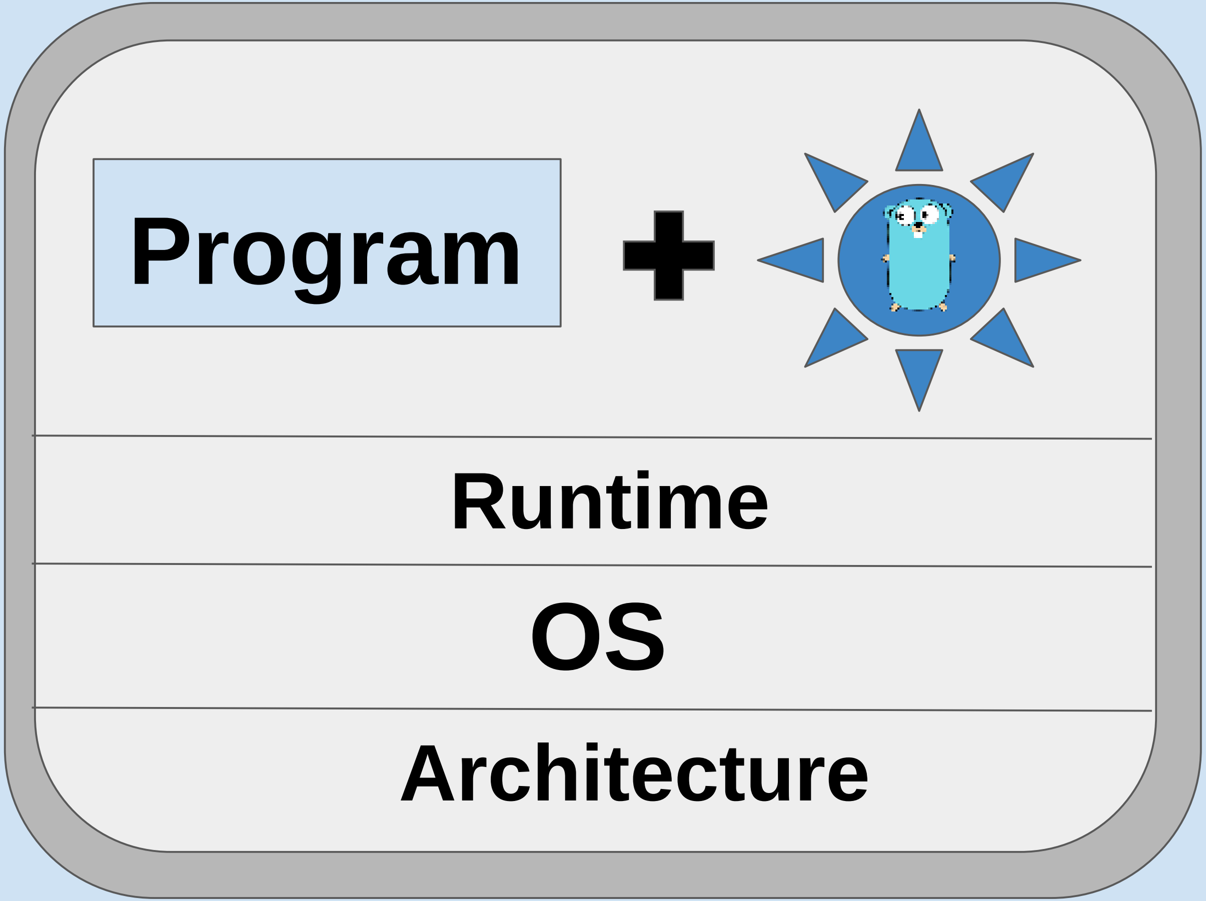
A Go Binary



What about multiple requests?

♫We don’t need no Multiple Instances♫

A Go Binary



Program ✚

Runtime

OS

Architecture

Coroutines & Green Threads

No one likes Threads.

Coroutines

Coroutines

Event Loop

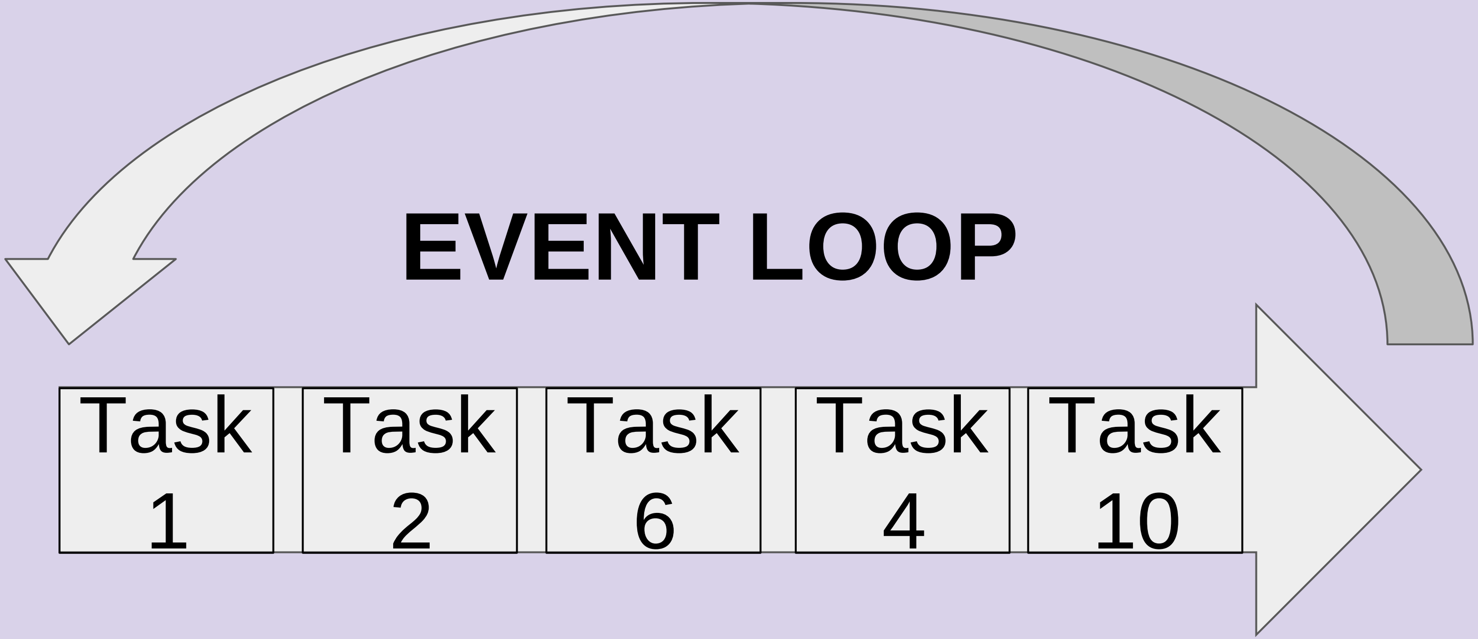
Components of a Coroutine

Life of a Coroutine

Coroutines

EVENT LOOP

Coroutines



EVENT LOOP

Task 1

Task 2

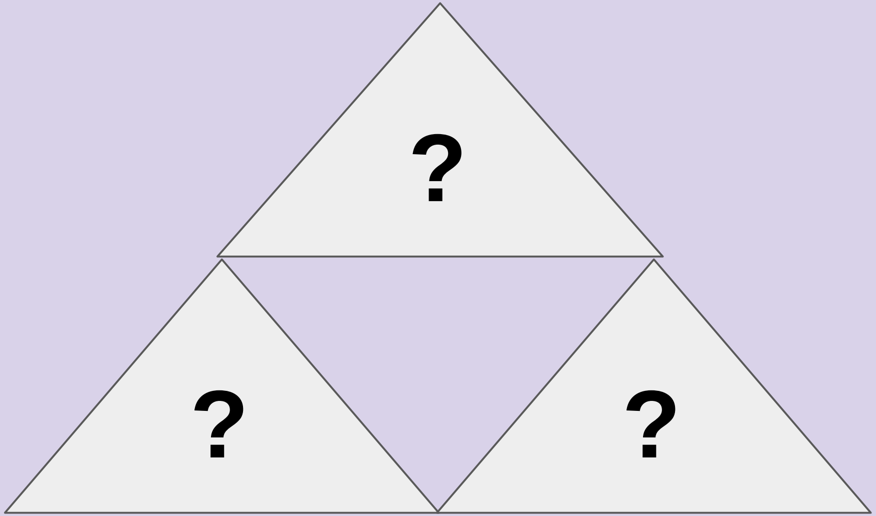
Task 6

Task 4

Task 10

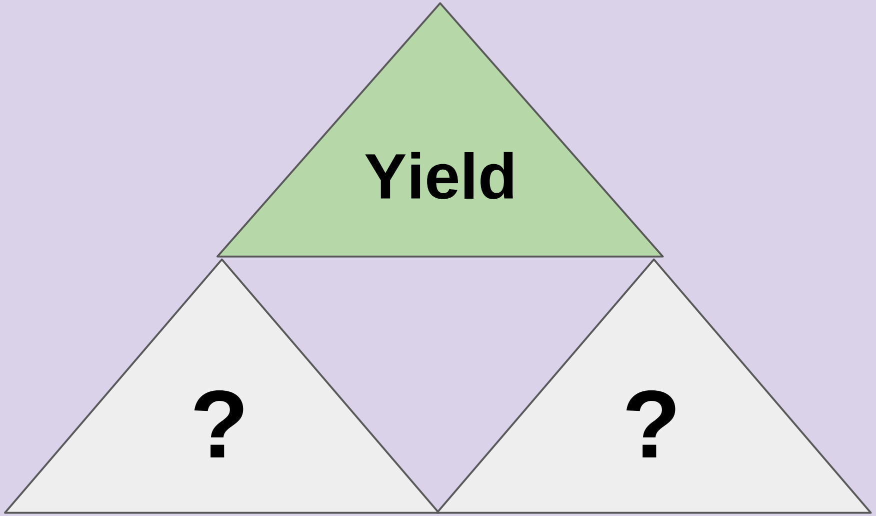
Components of Coroutine

Coroutines



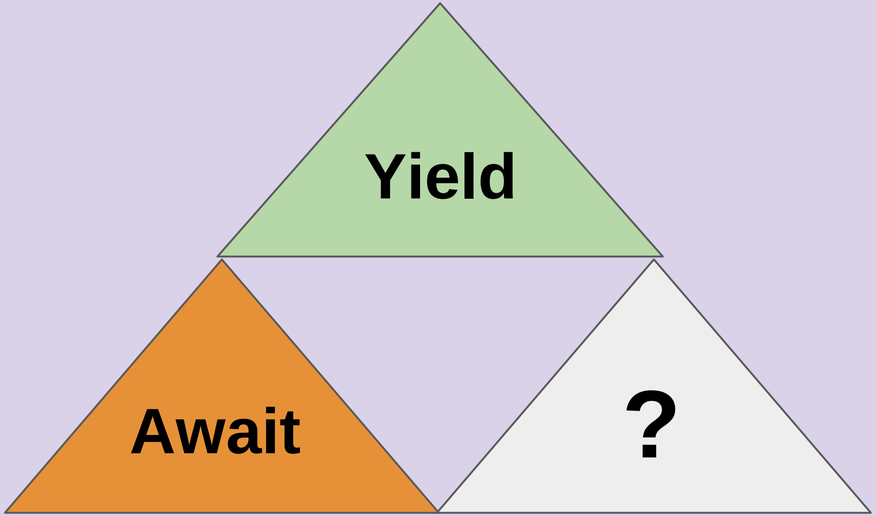
Components of Coroutine

Coroutines



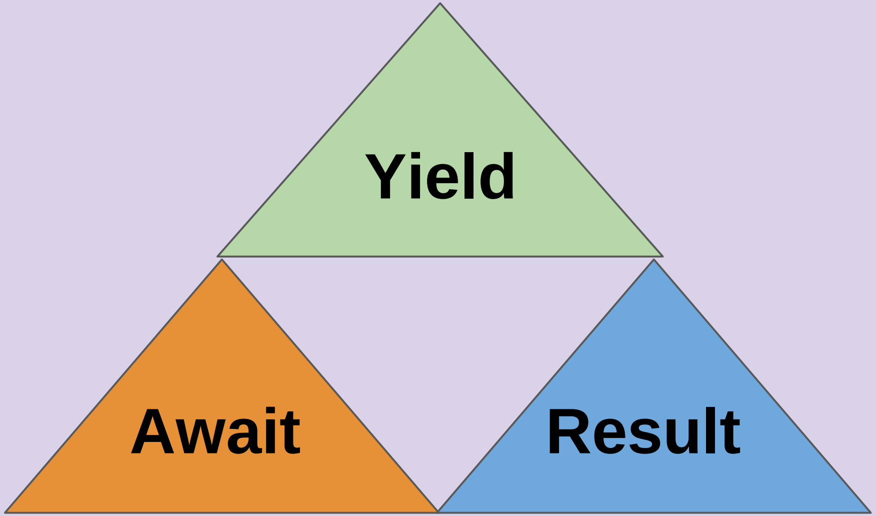
Components of Coroutine

Coroutines



Components of Coroutine

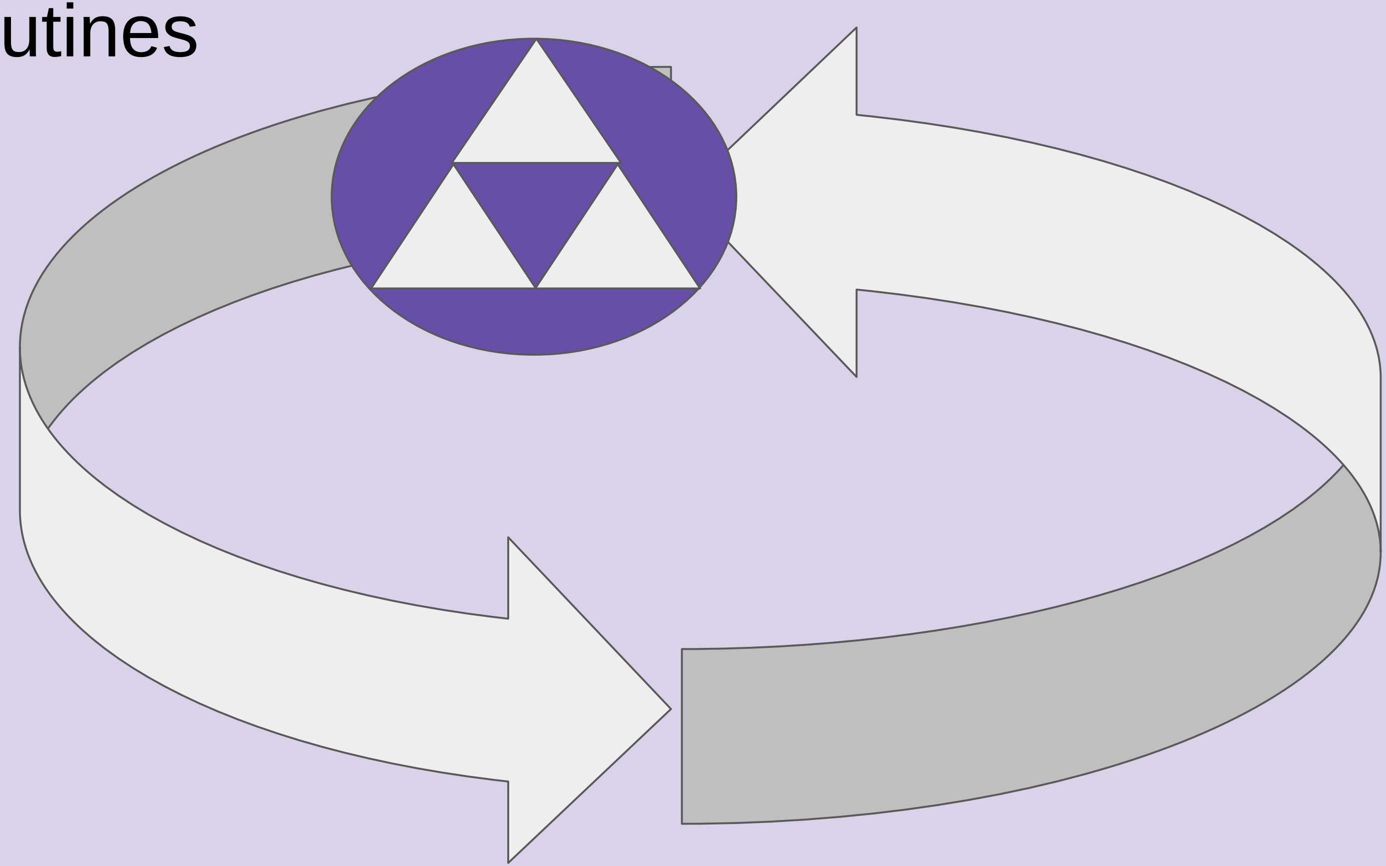
Coroutines



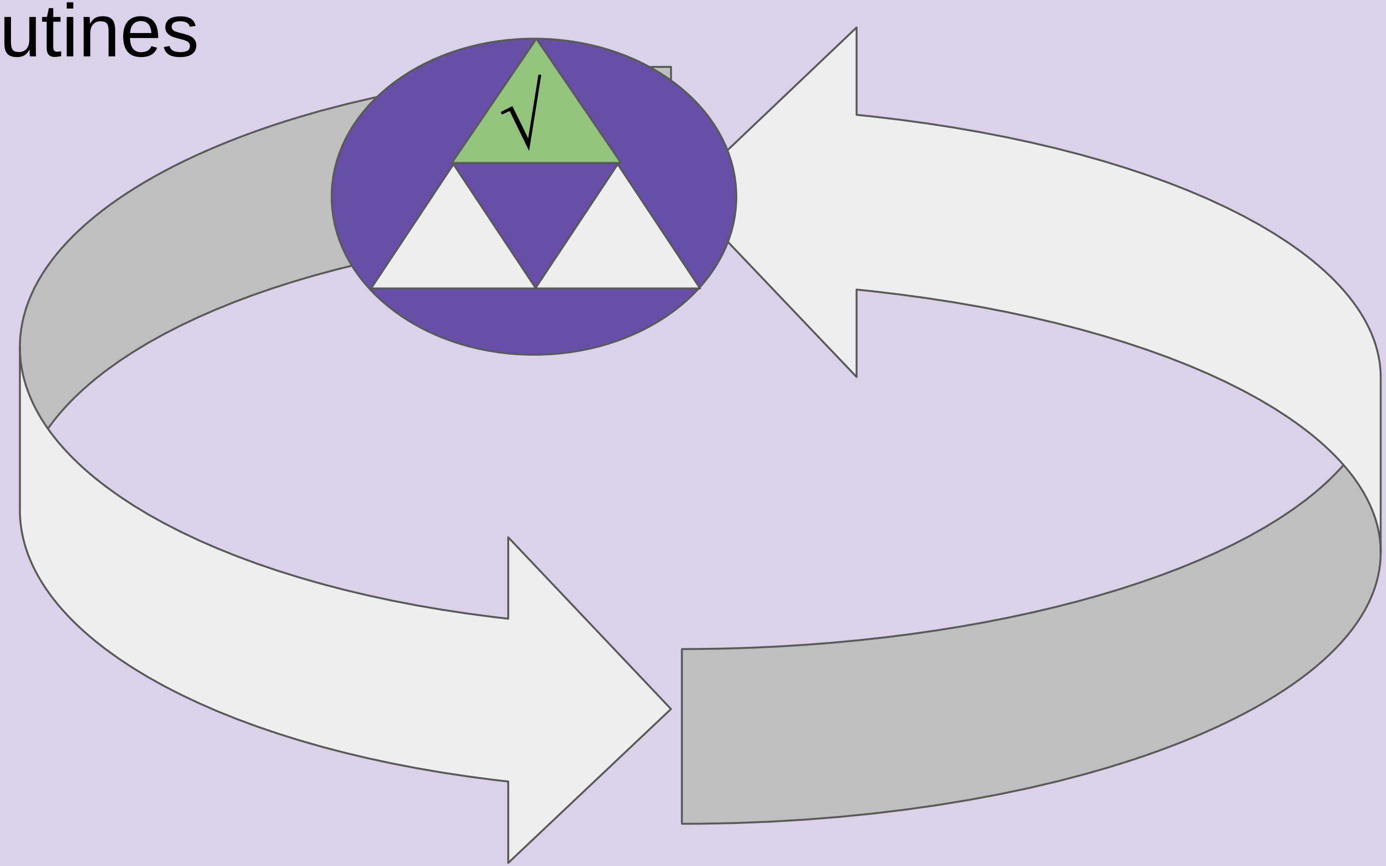
Components of Coroutine

Life of a Coroutine

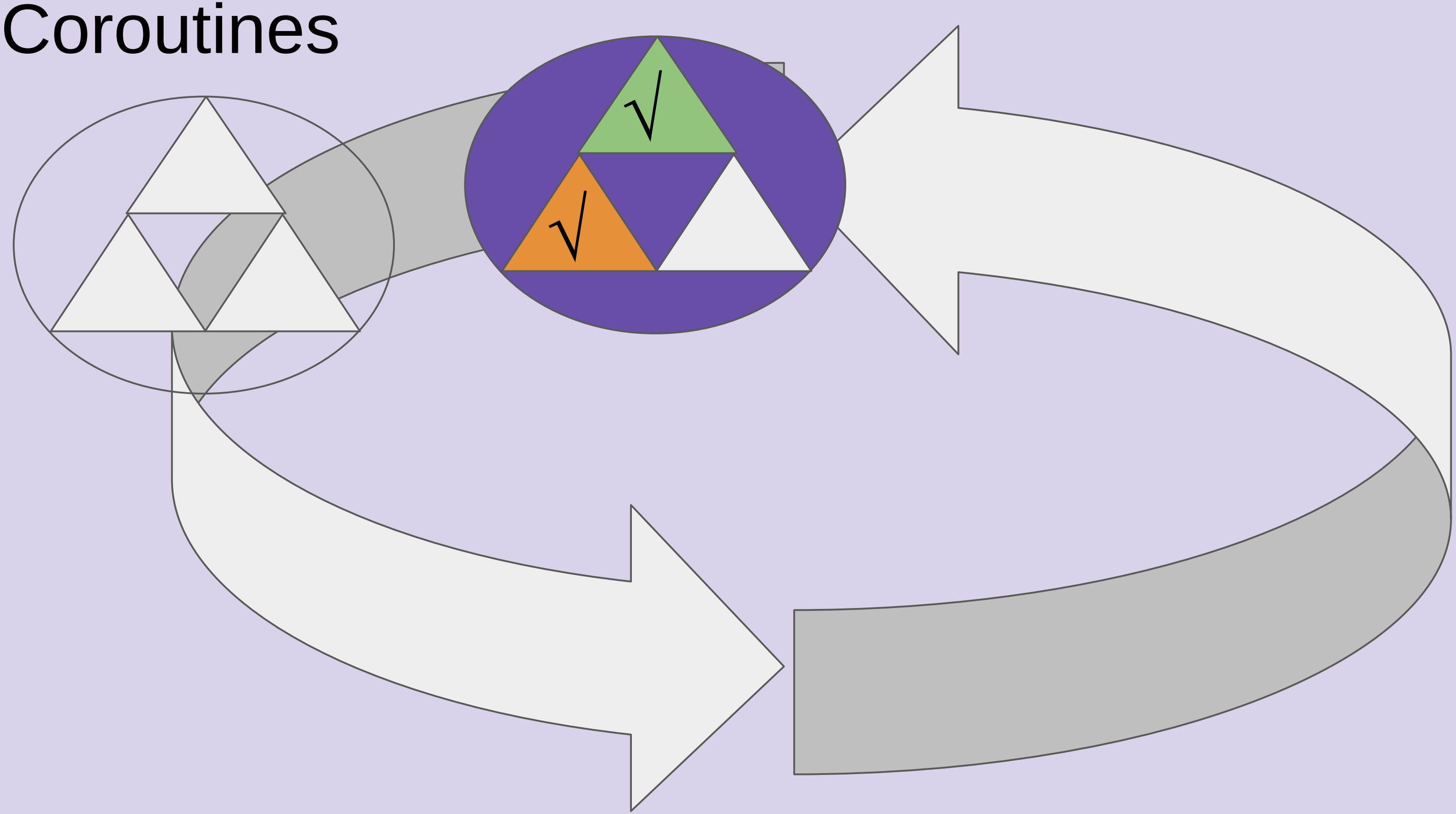
Coroutines



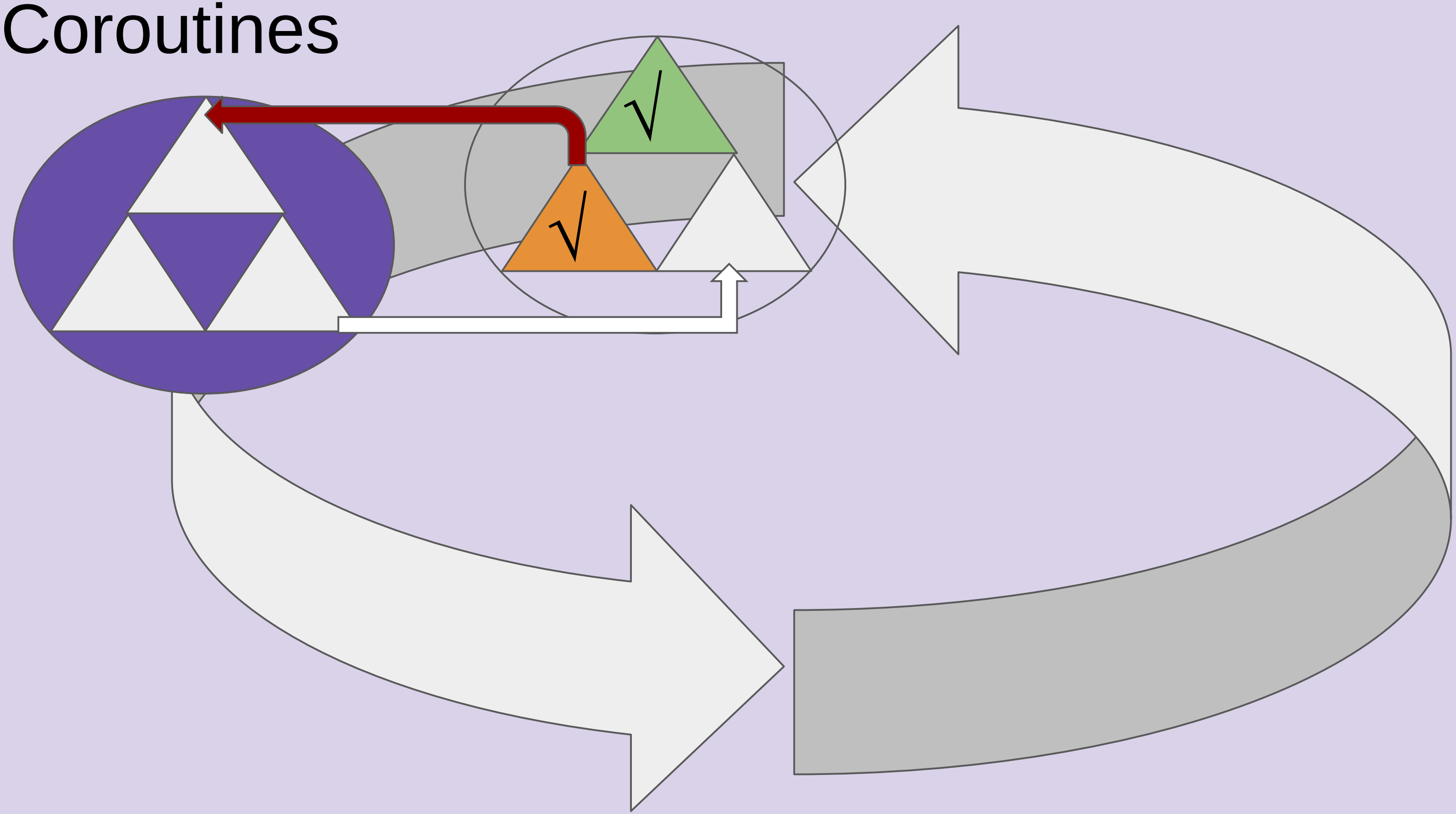
Coroutines



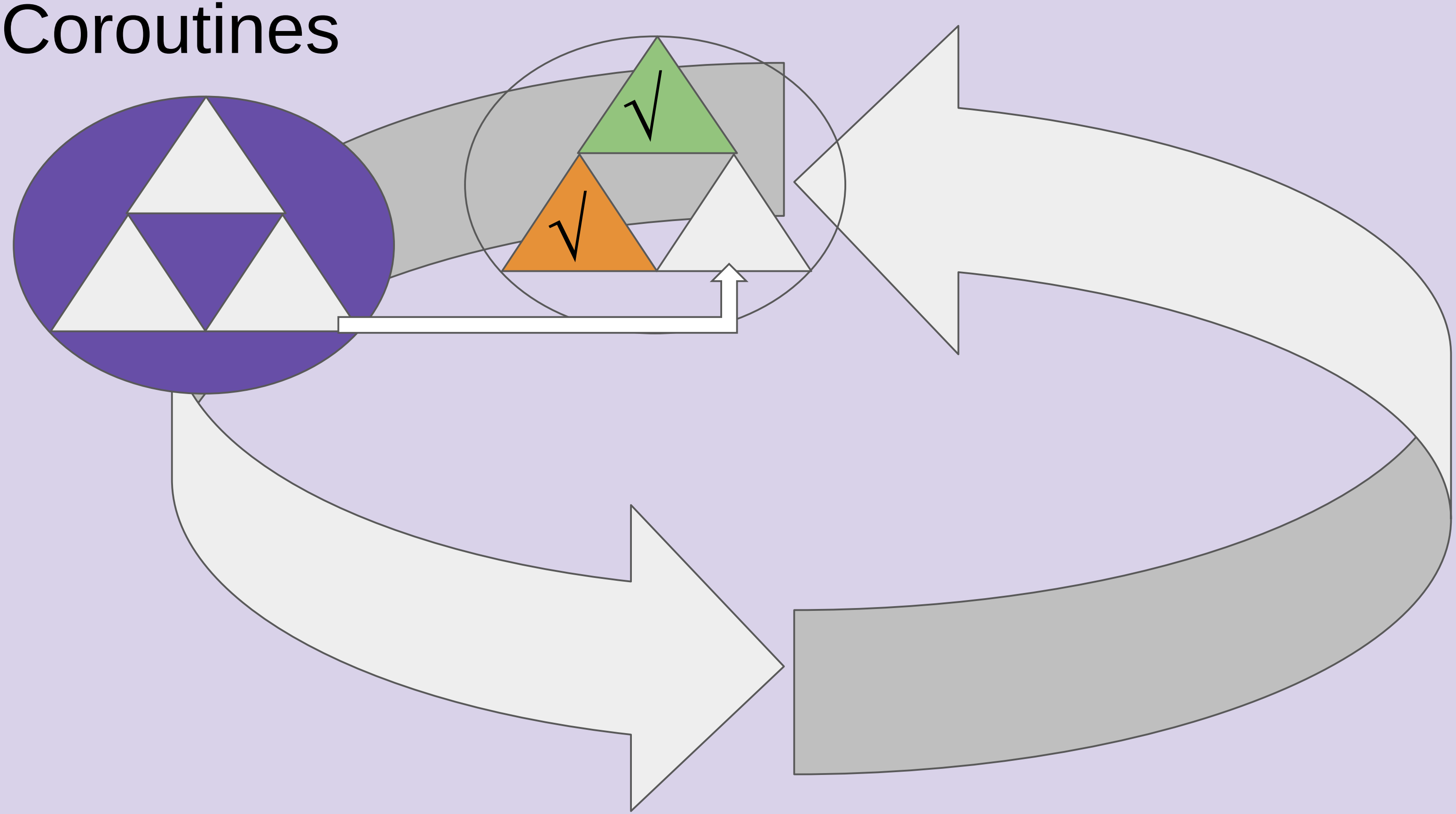
Coroutines



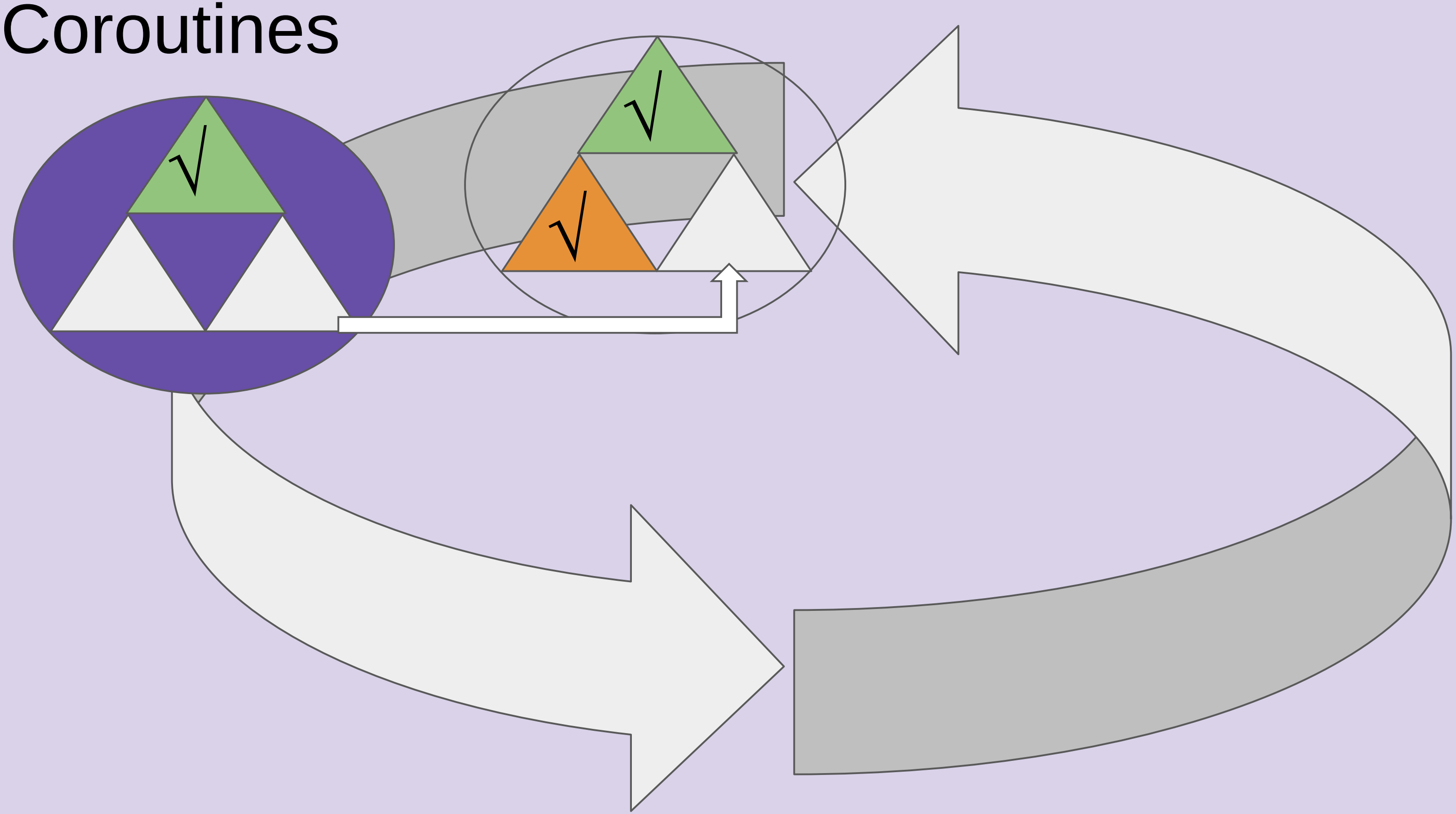
Coroutines



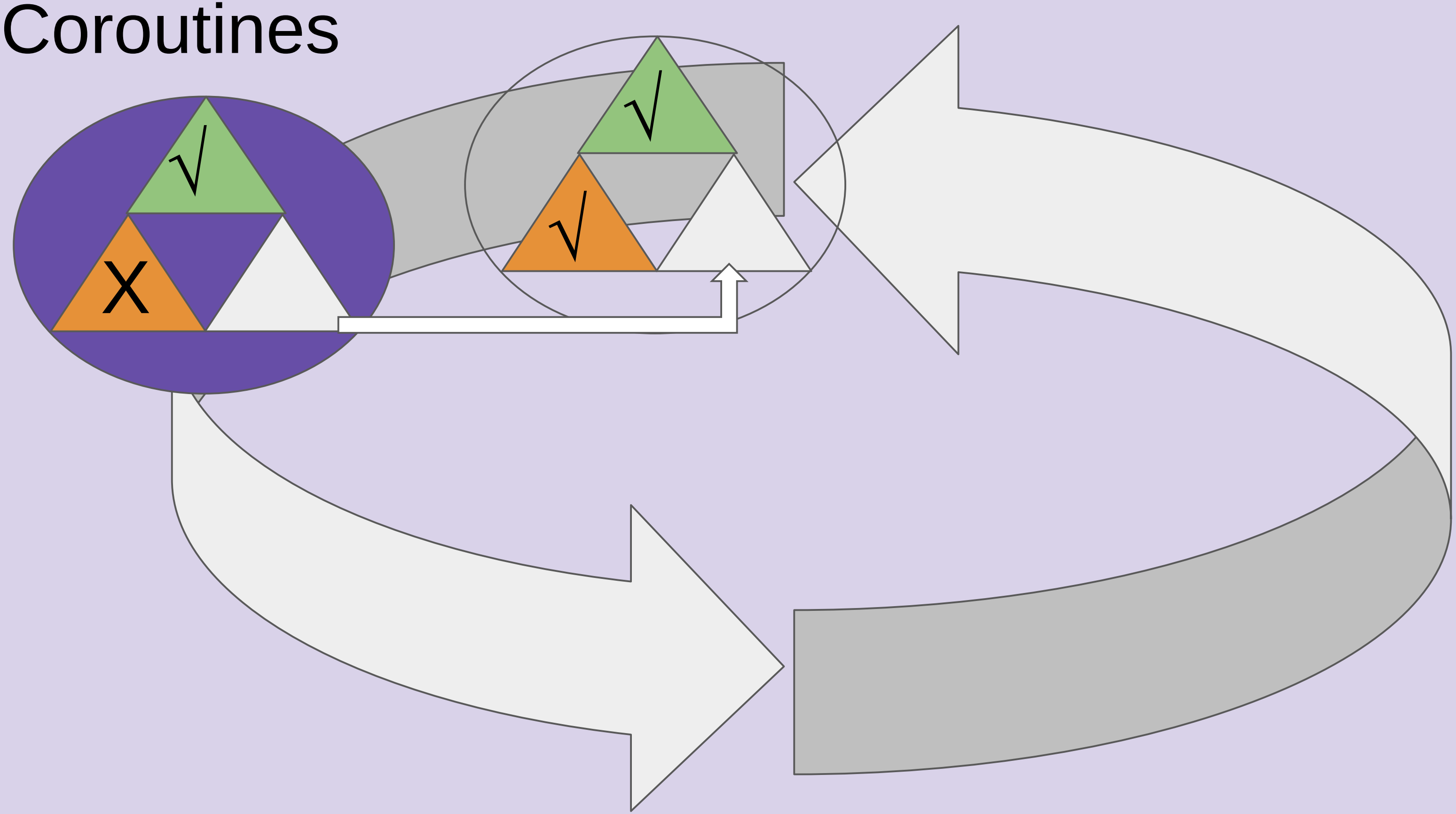
Coroutines



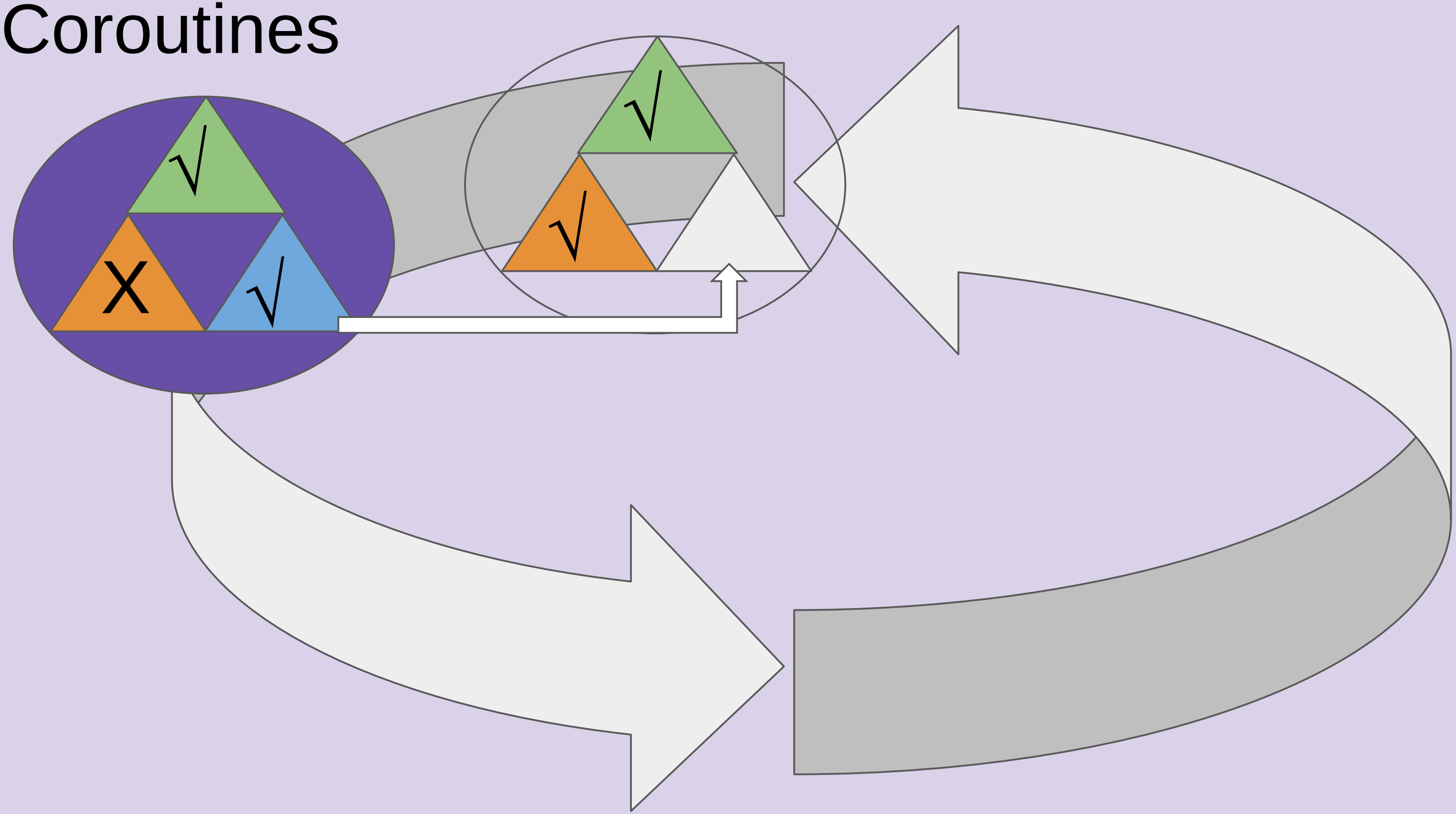
Coroutines



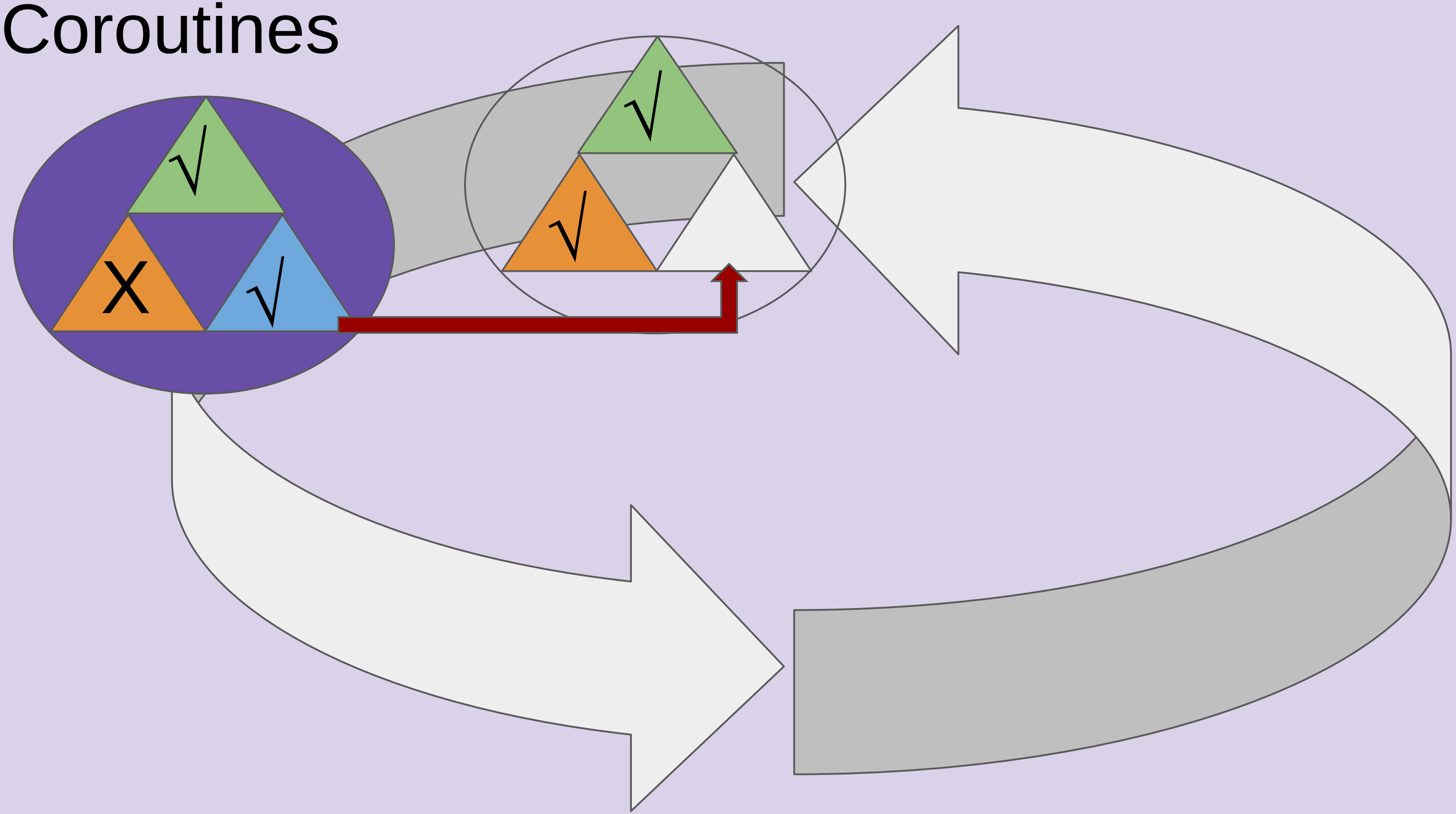
Coroutines



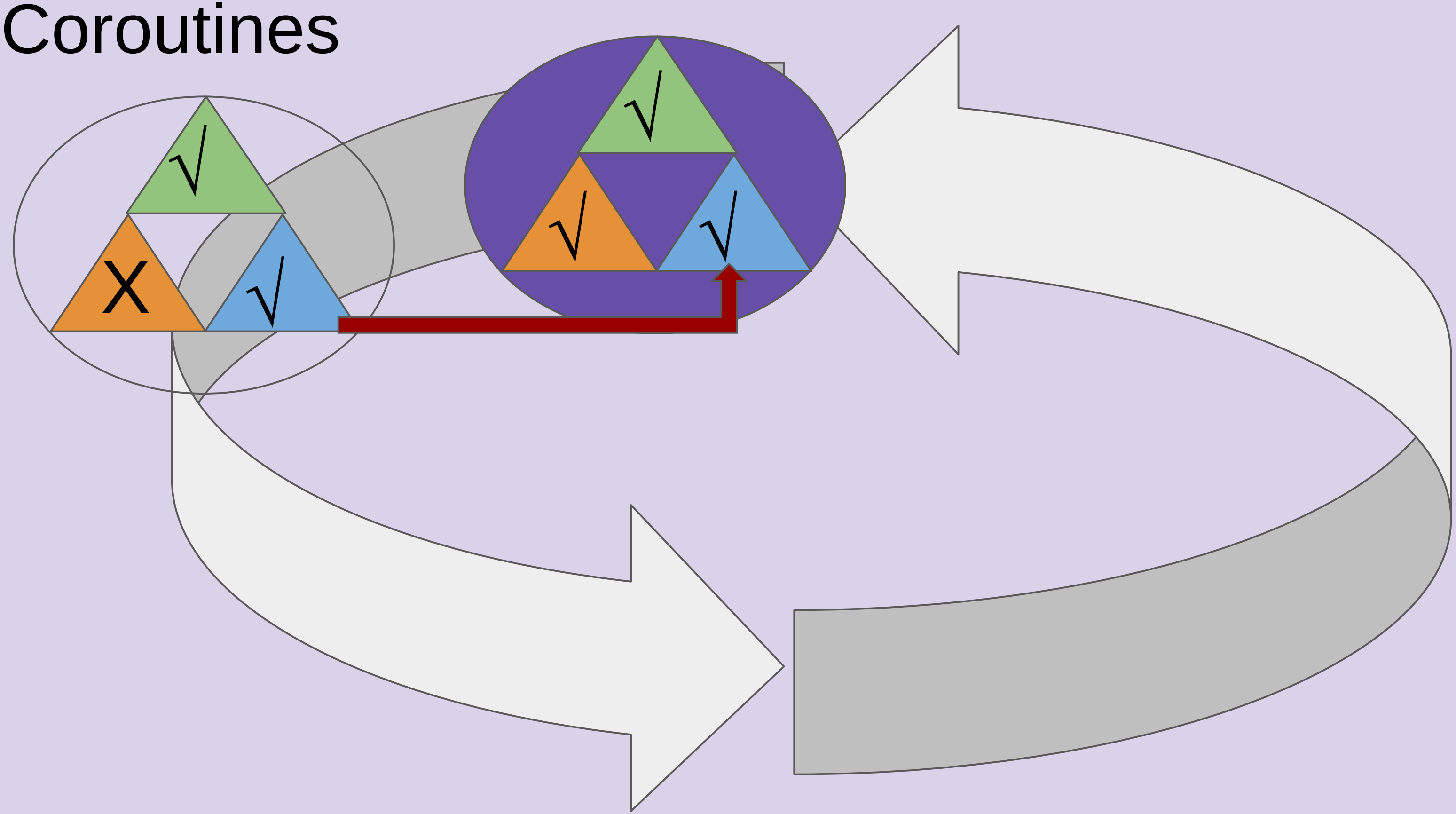
Coroutines



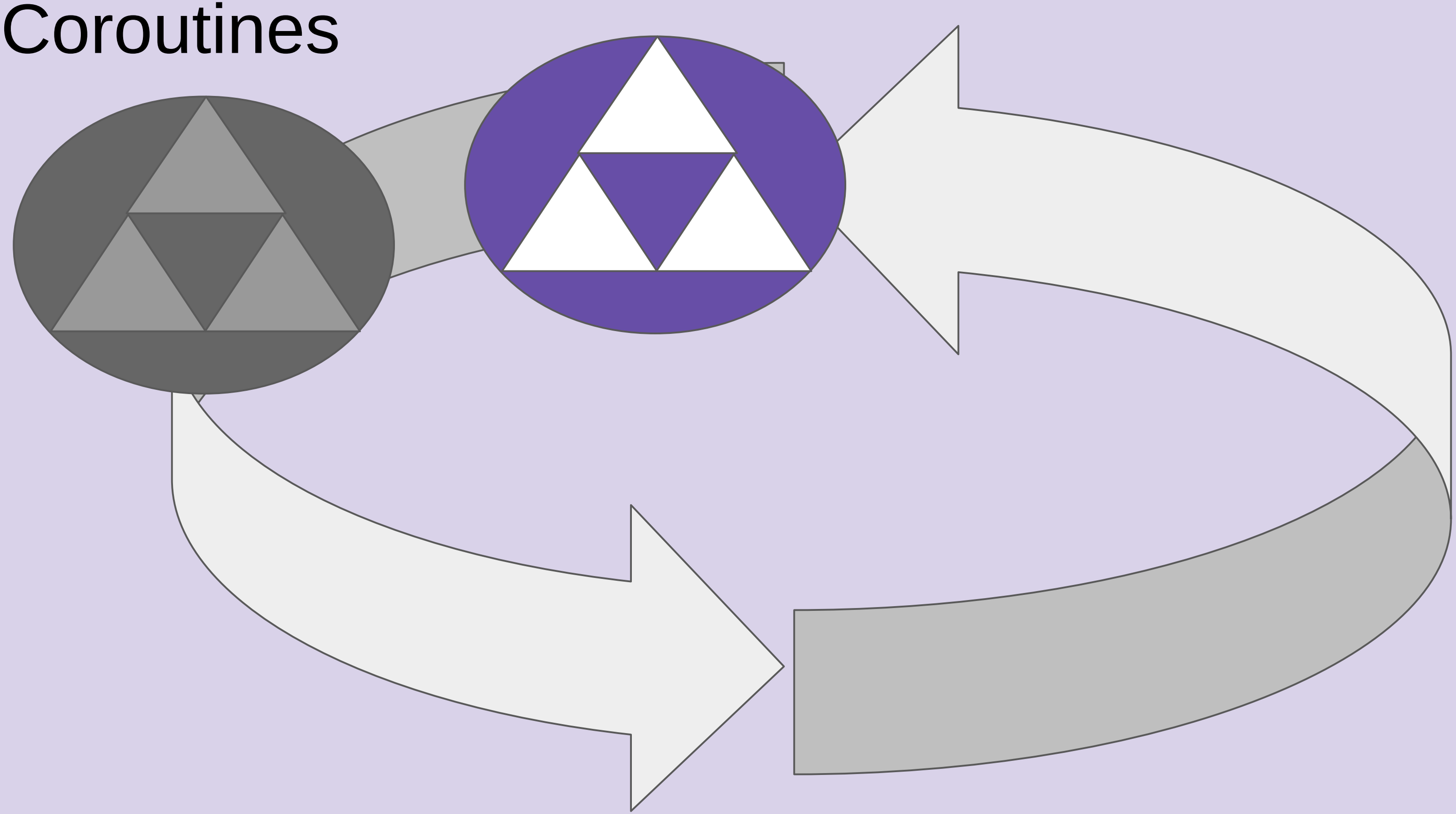
Coroutines



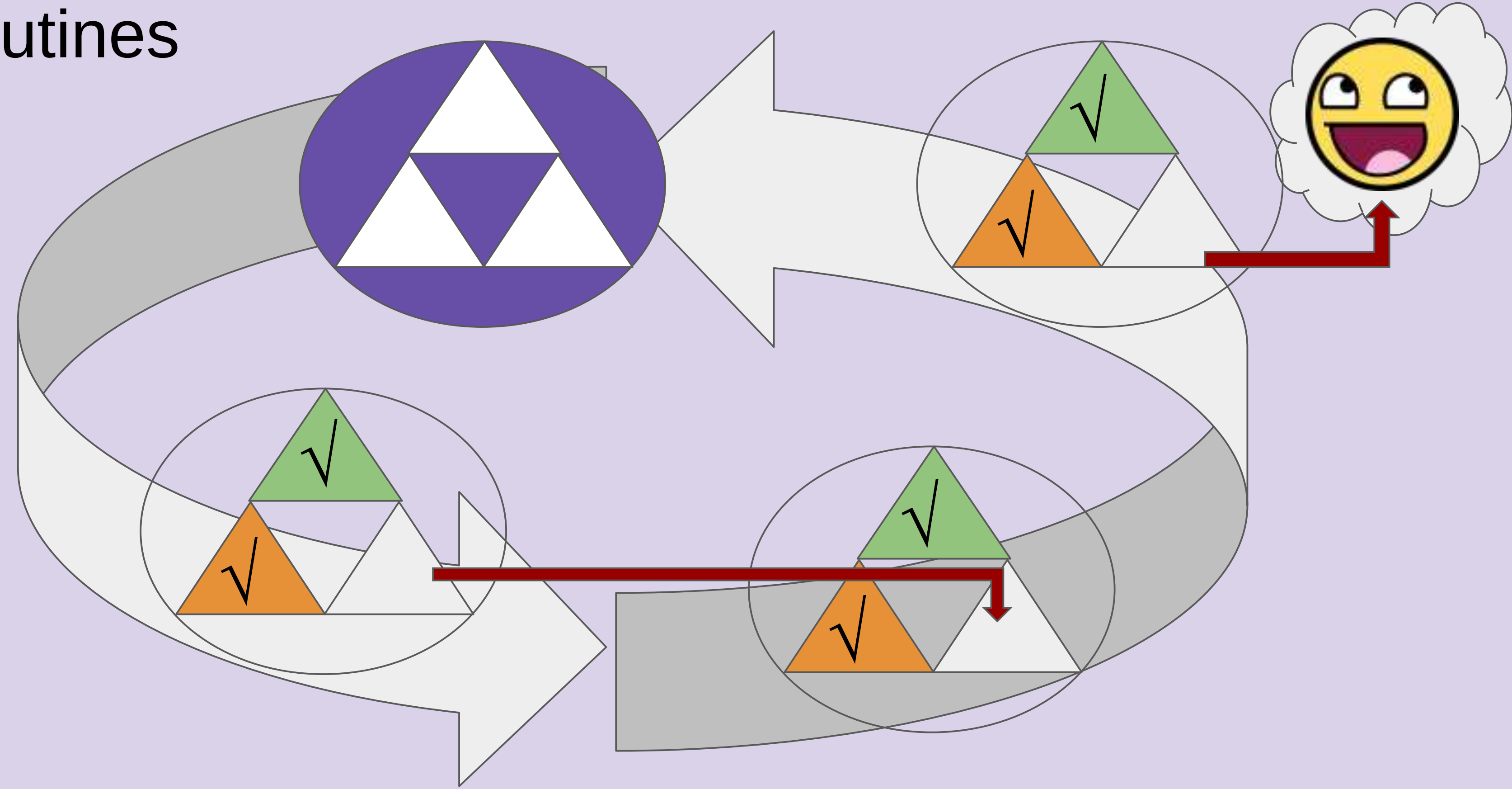
Coroutines



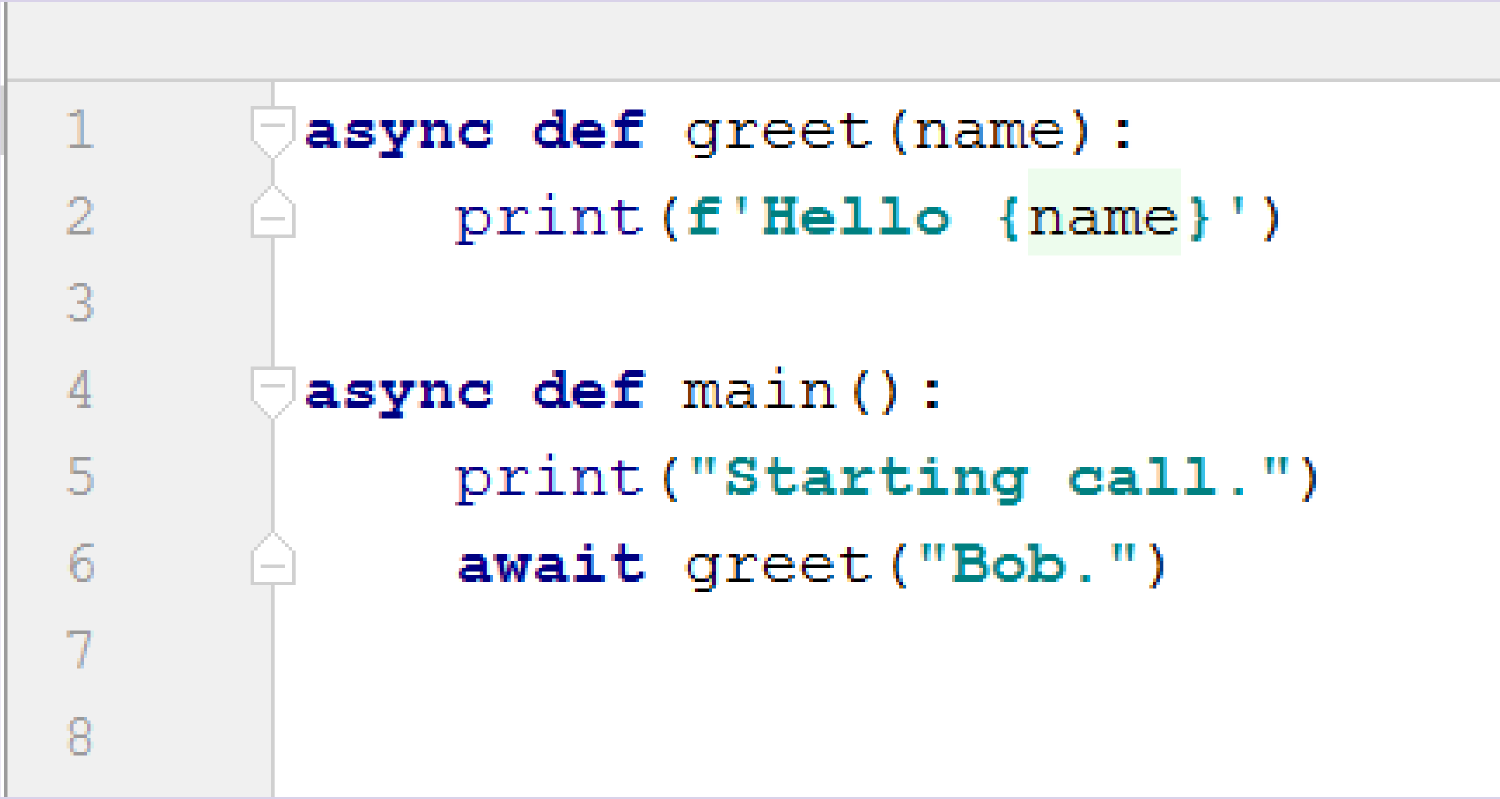
Coroutines



Coroutines



Coroutines - Python

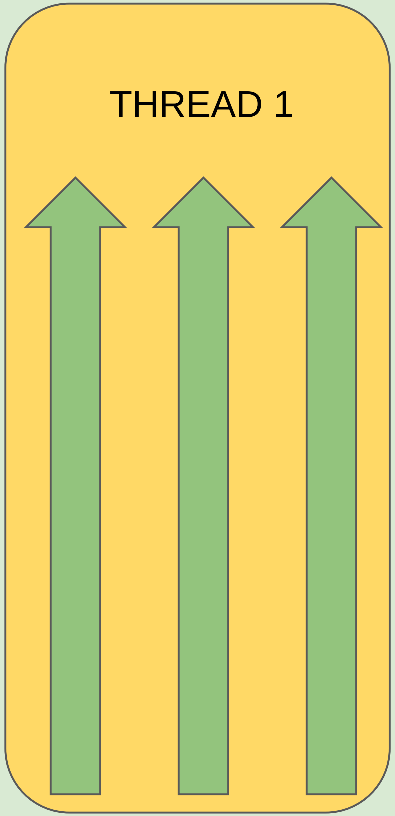


```python
async def greet(name):
    print(f'Hello {name}')

async def main():
    print("Starting call.")
    await greet("Bob.")
```

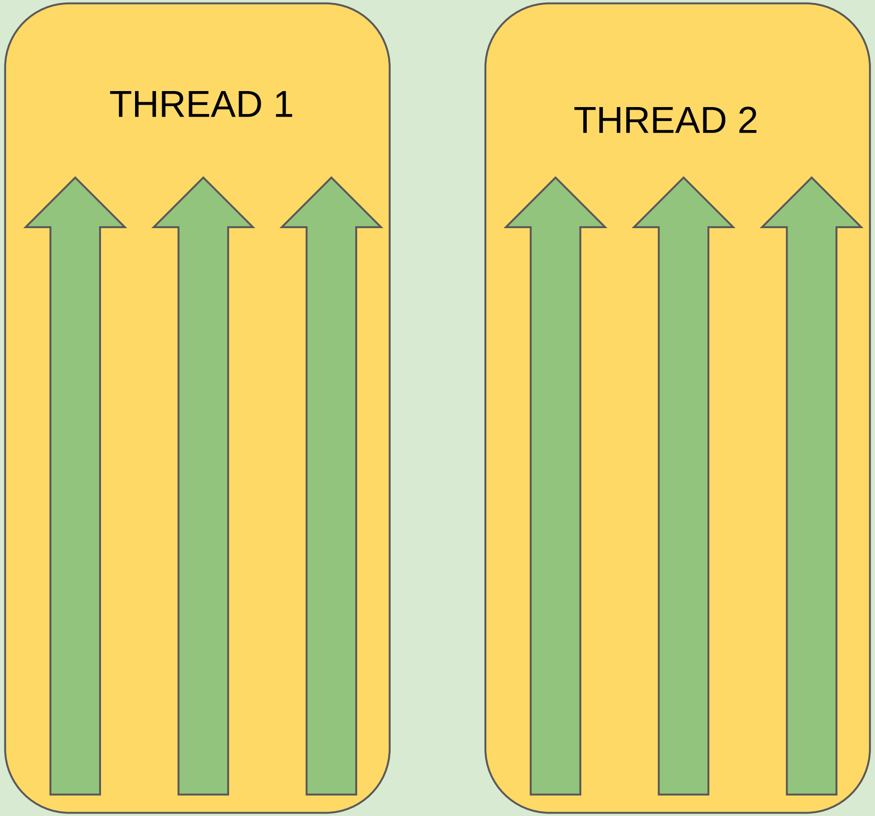
Green Threads

They are a lot like Coroutines

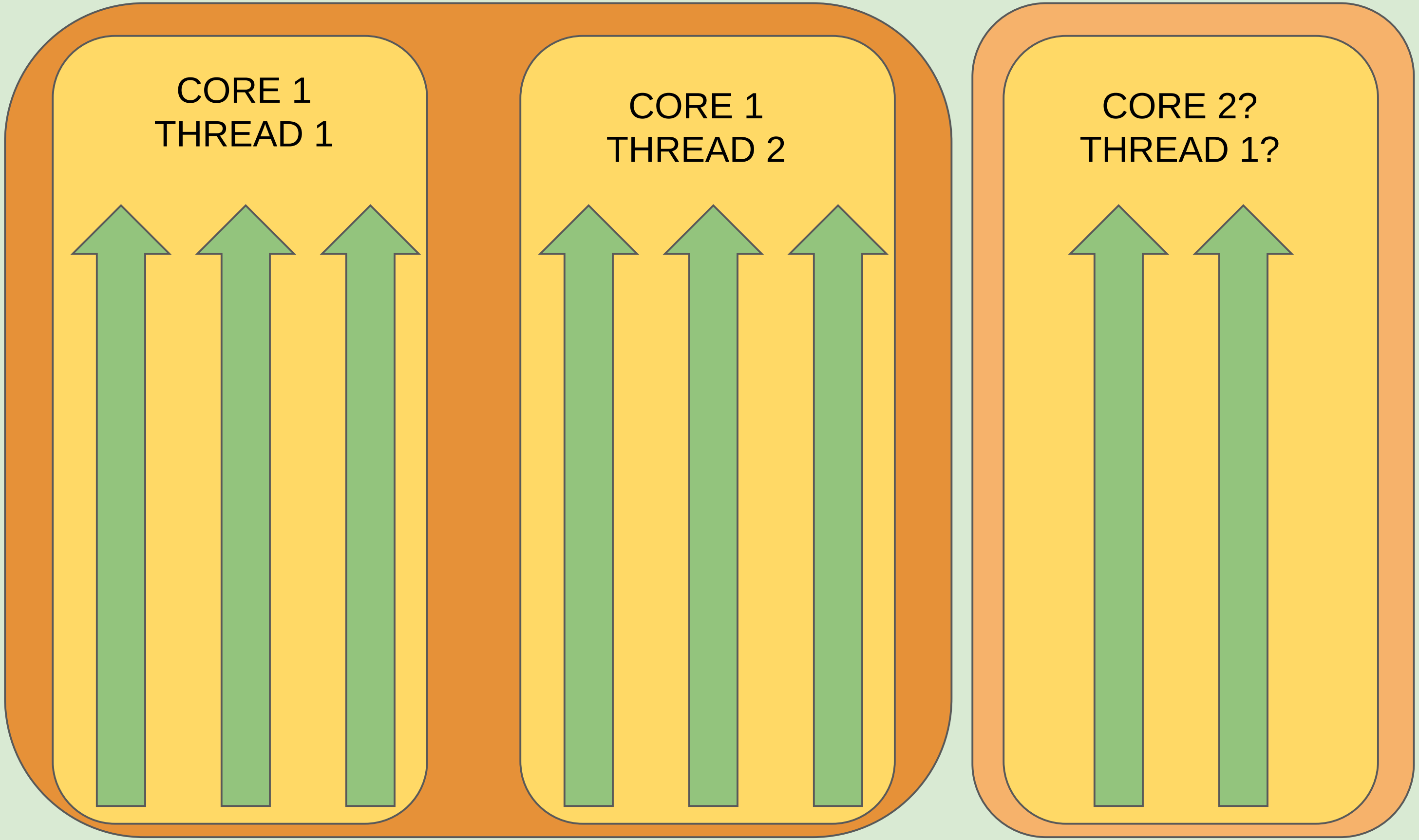
Green Threads



Green Threads



Green Threads

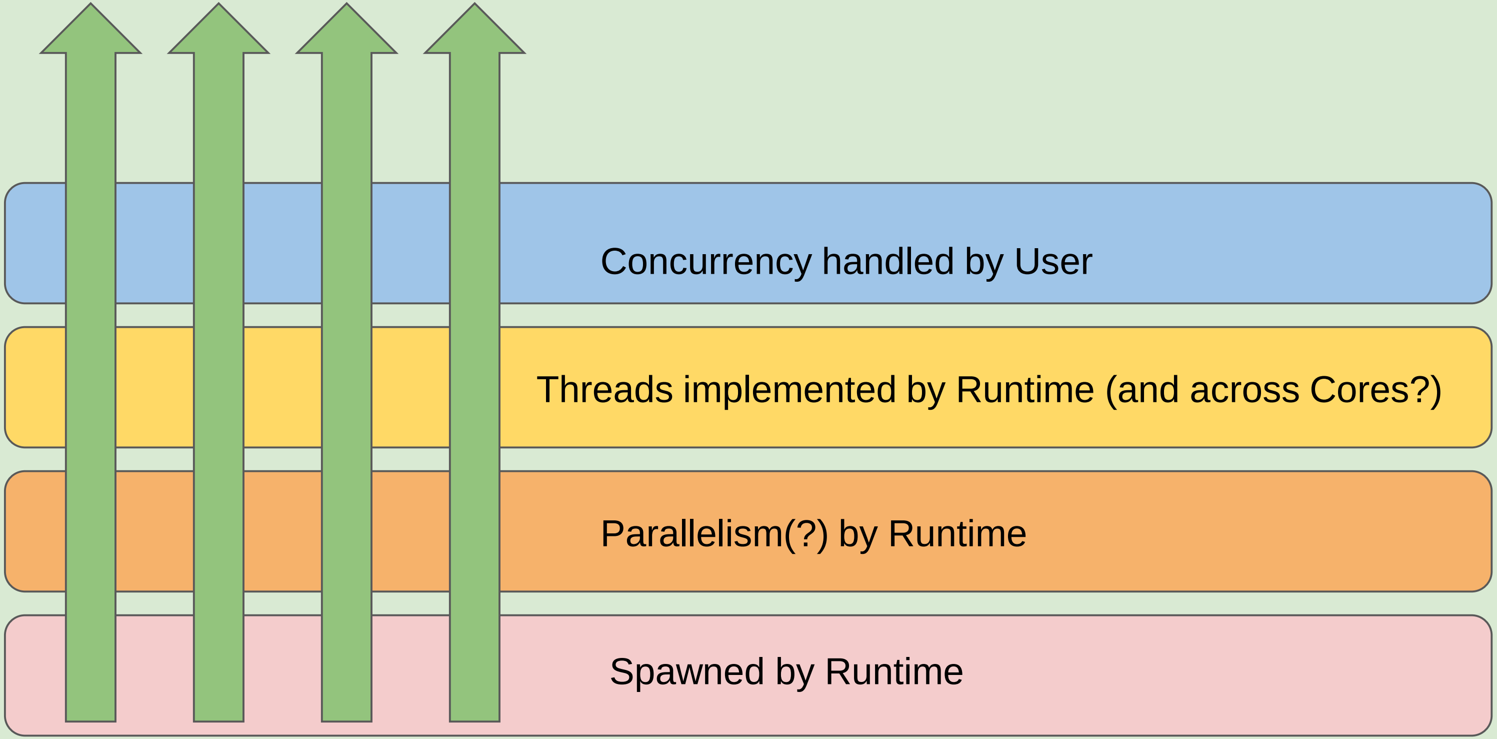


CORE 1
THREAD 1

CORE 1
THREAD 2

CORE 2?
THREAD 1?

Green Threads



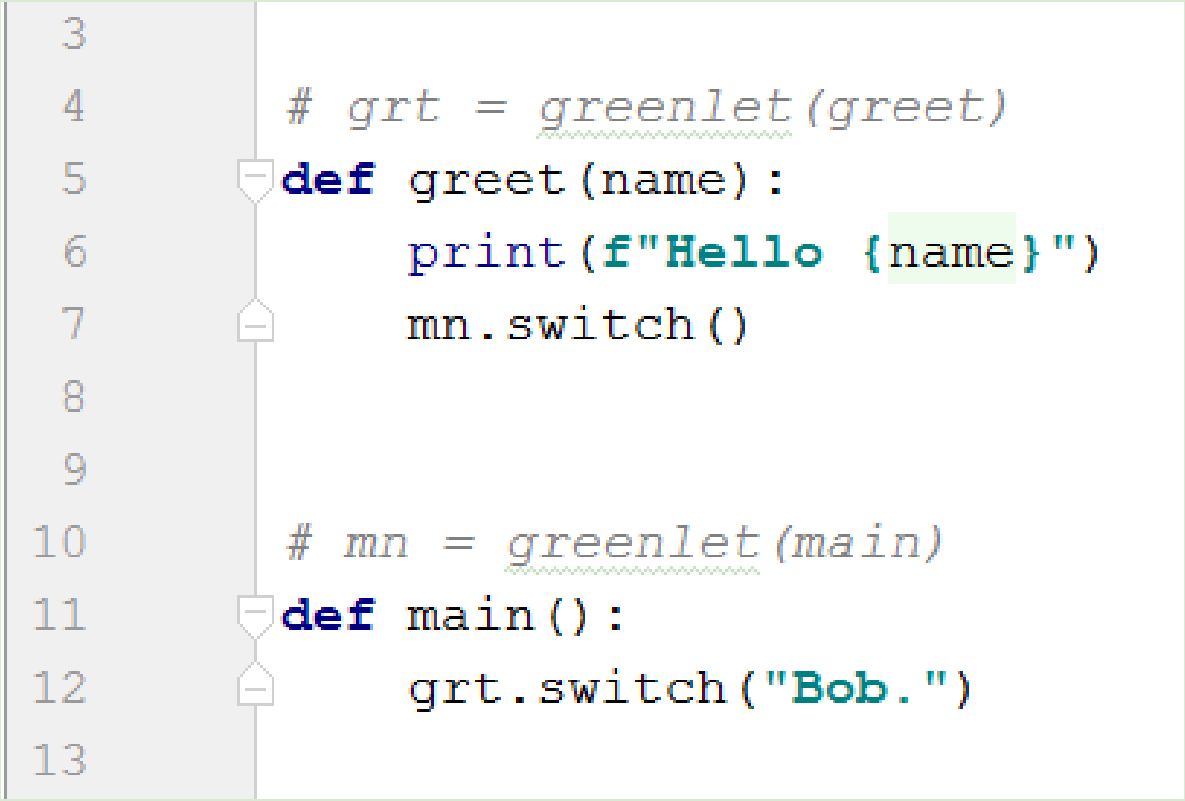
Concurrency handled by User

Threads implemented by Runtime (and across Cores?)

Parallelism(?) by Runtime

Spawned by Runtime

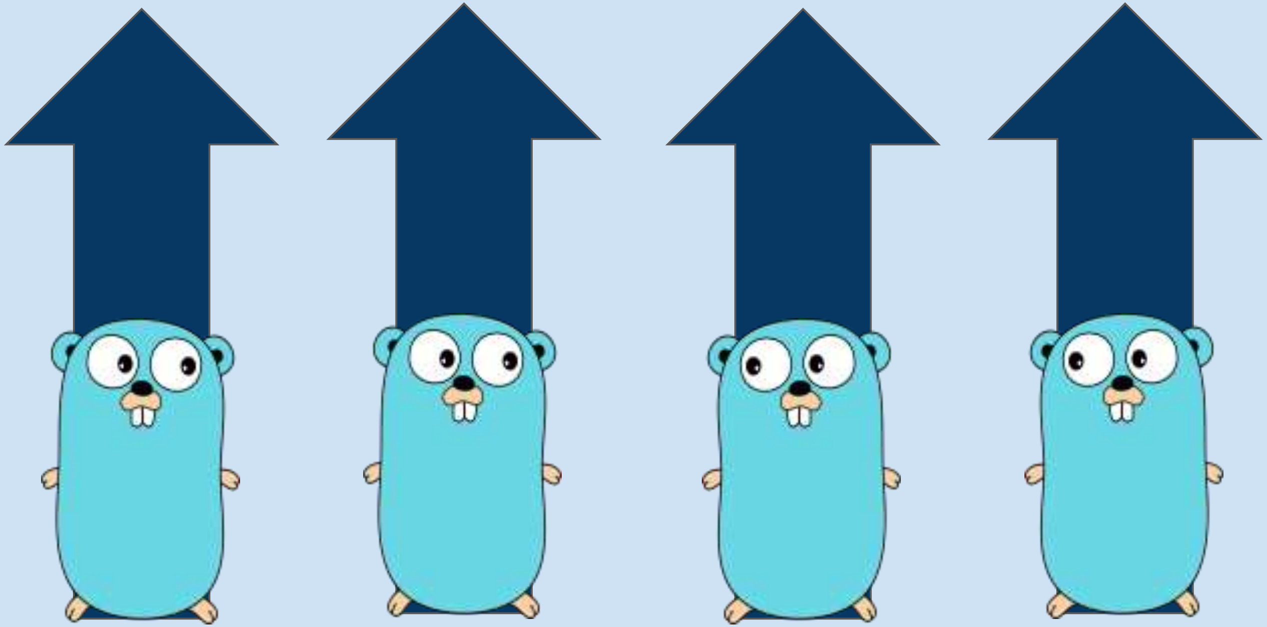
Green Threads - Python (greenlet)



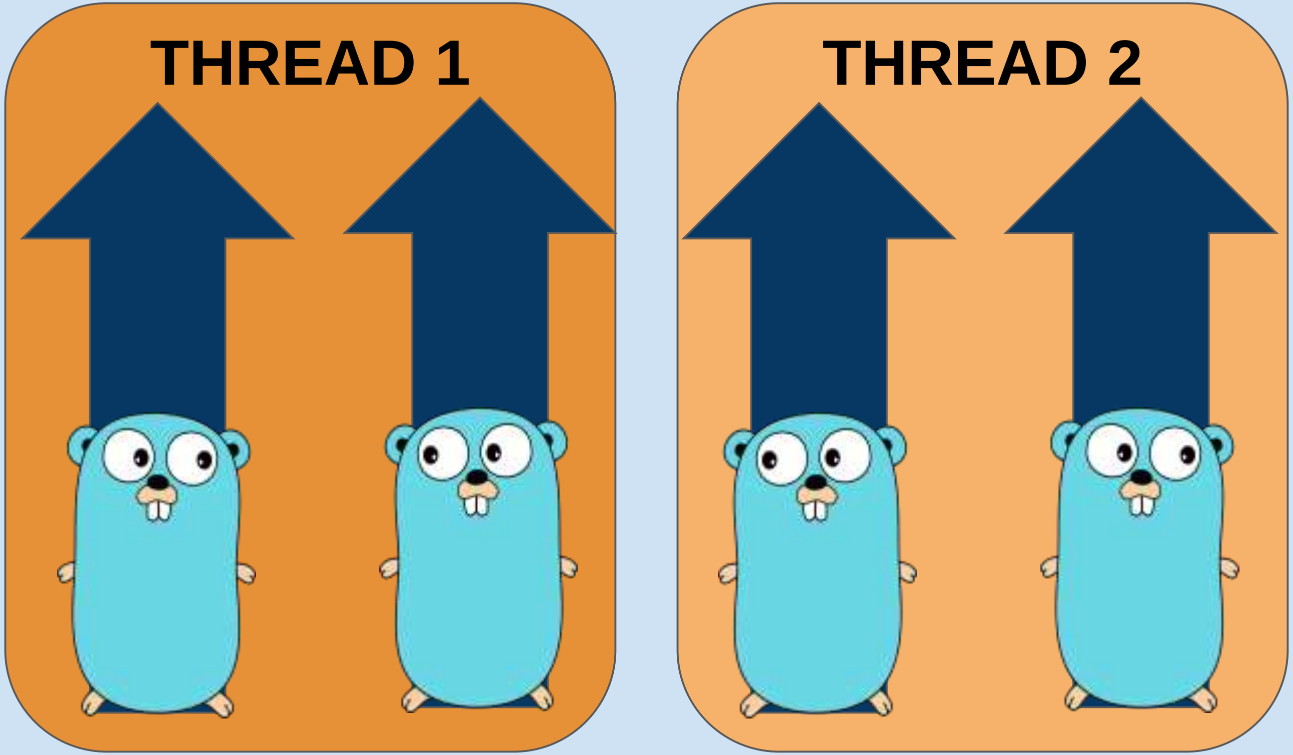
```

# grt = greenlet(greet)
def greet(name):
    print(f"Hello {name}")
    mn.switch()


# mn = greenlet(main)
def main():
    grt.switch("Bob.")

```

Goroutines

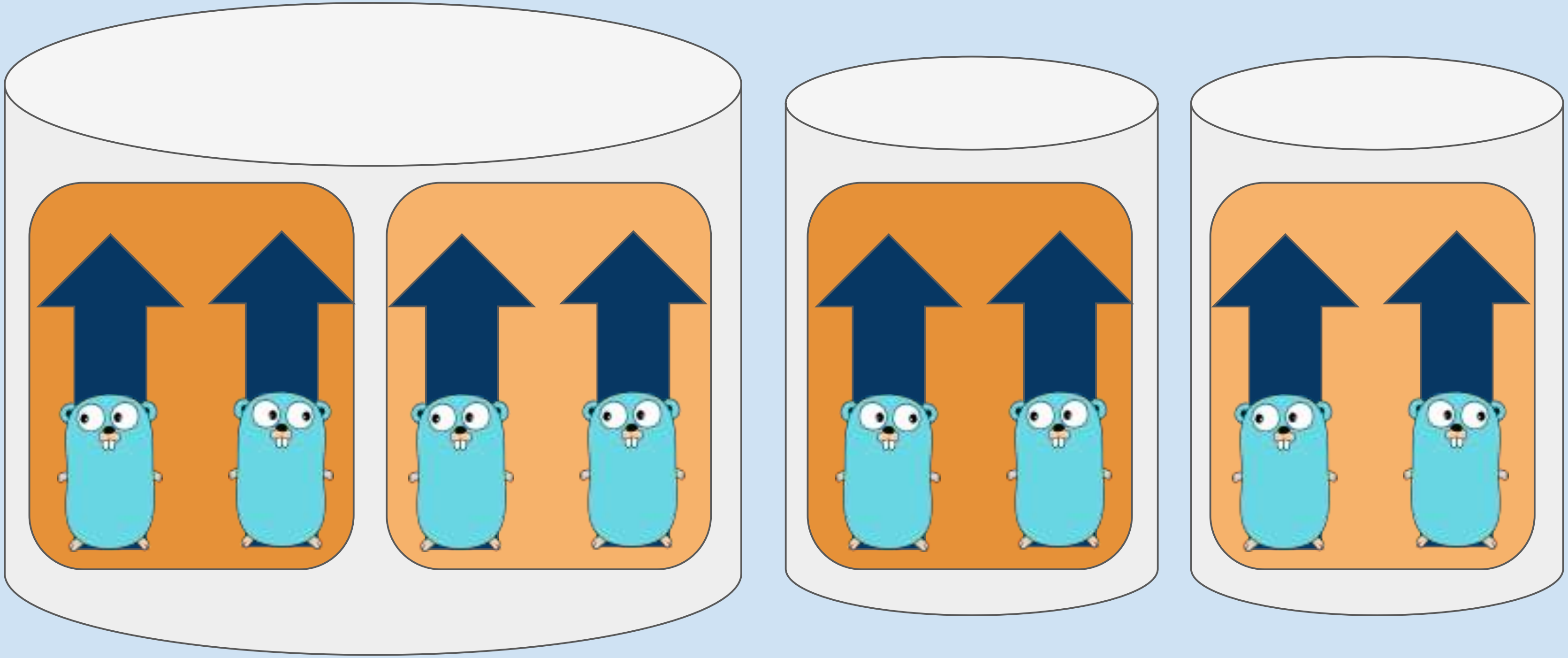
Goroutines



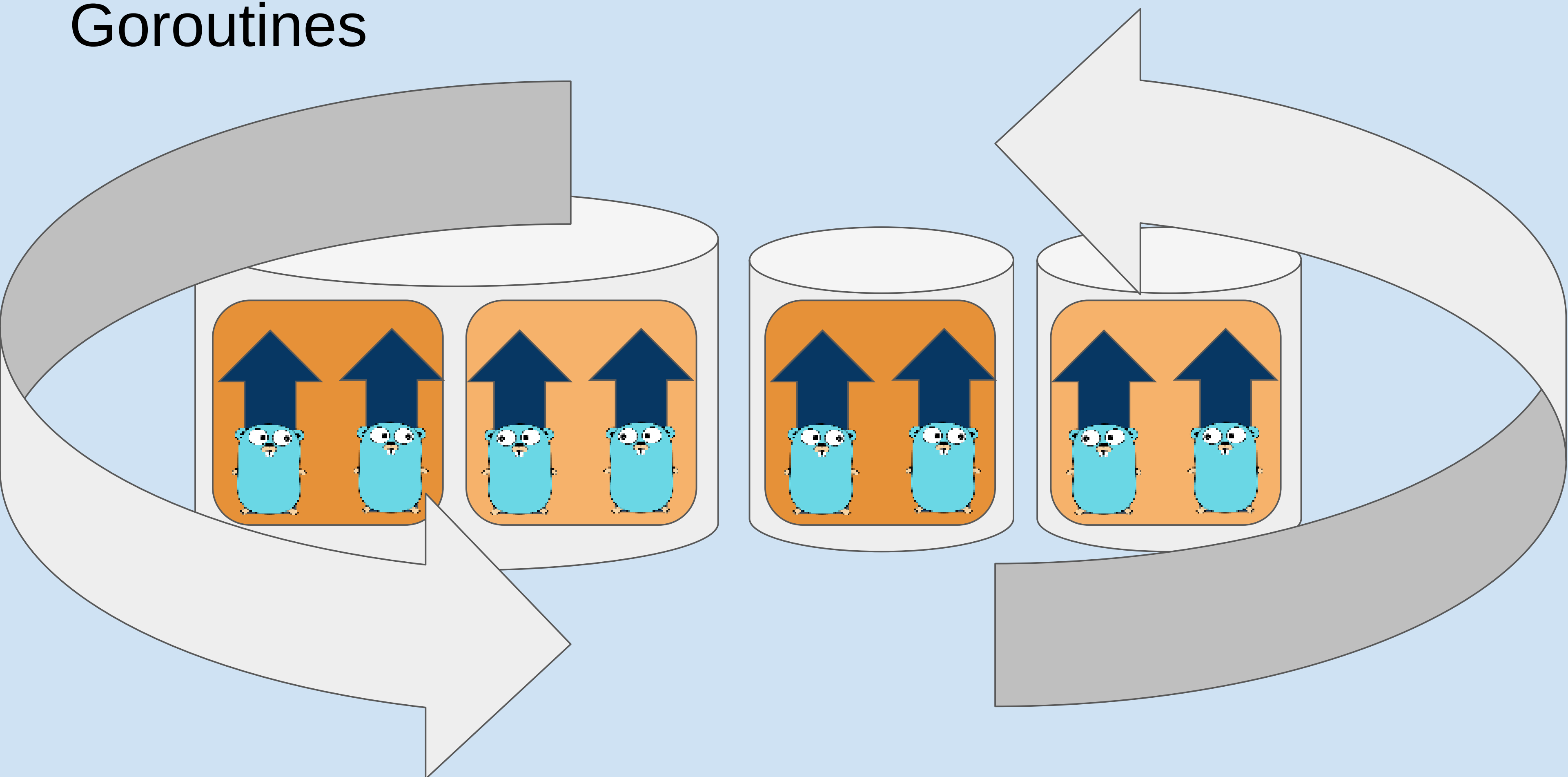
Goroutines



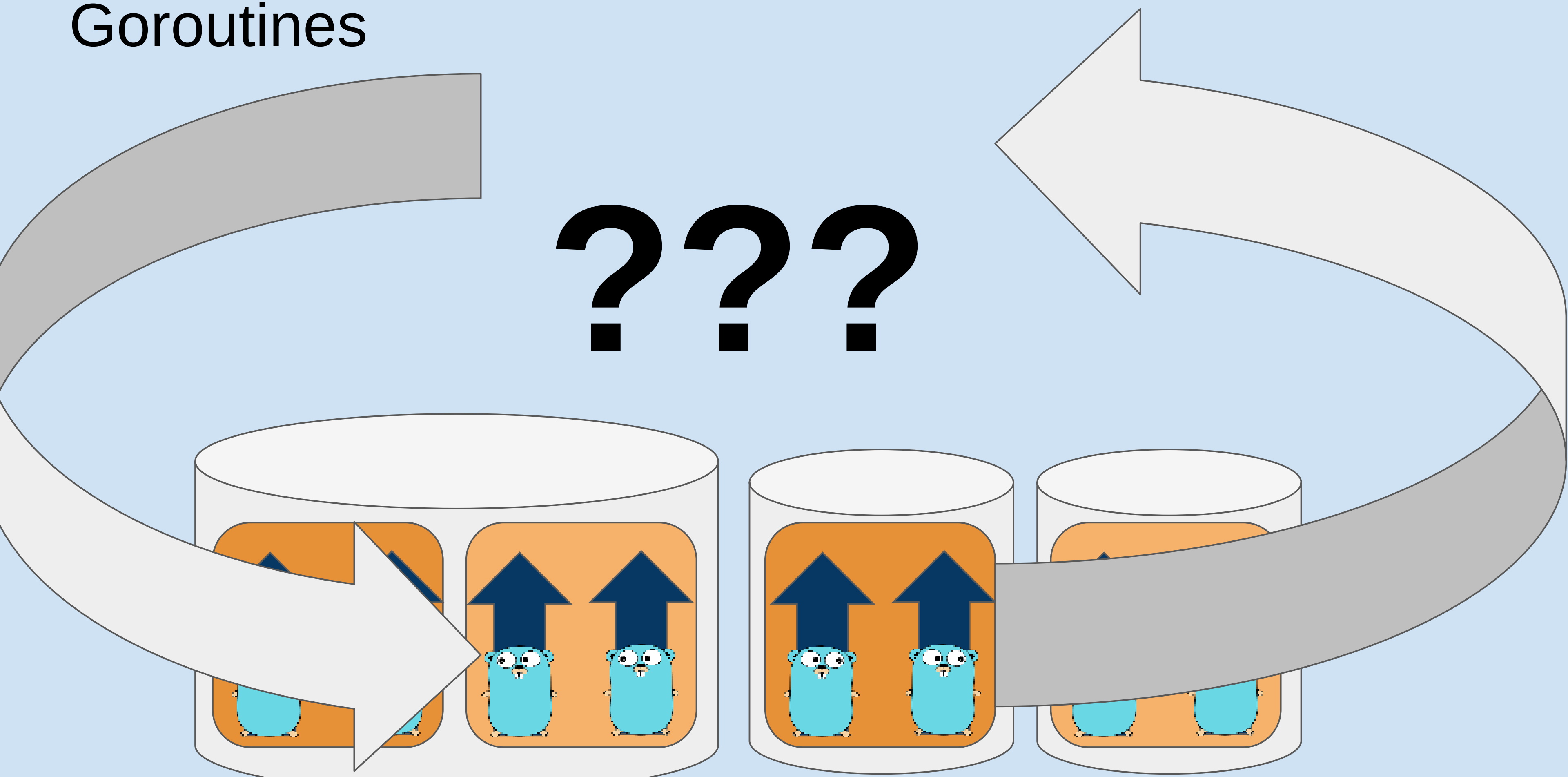
Goroutines



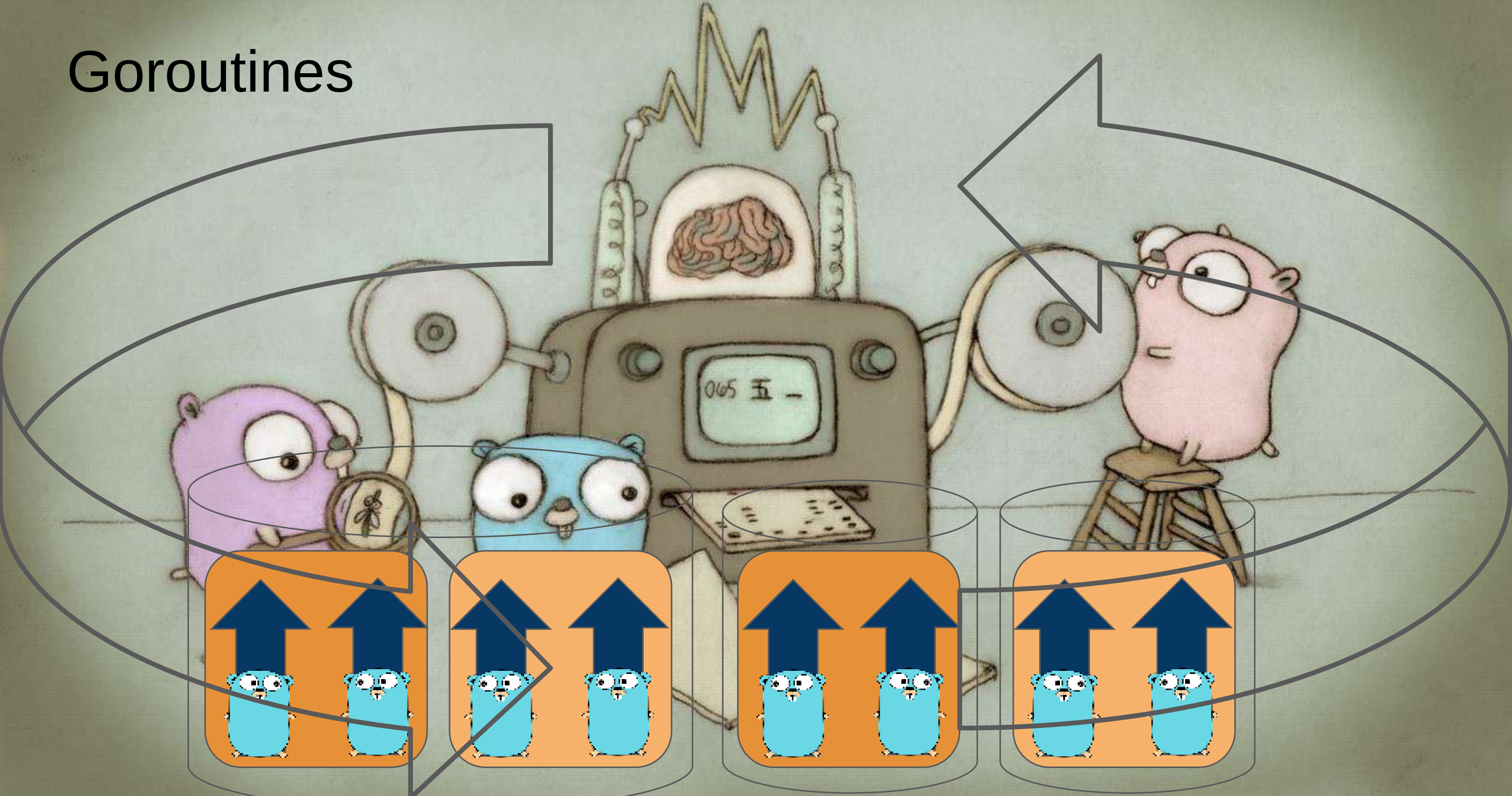
Goroutines



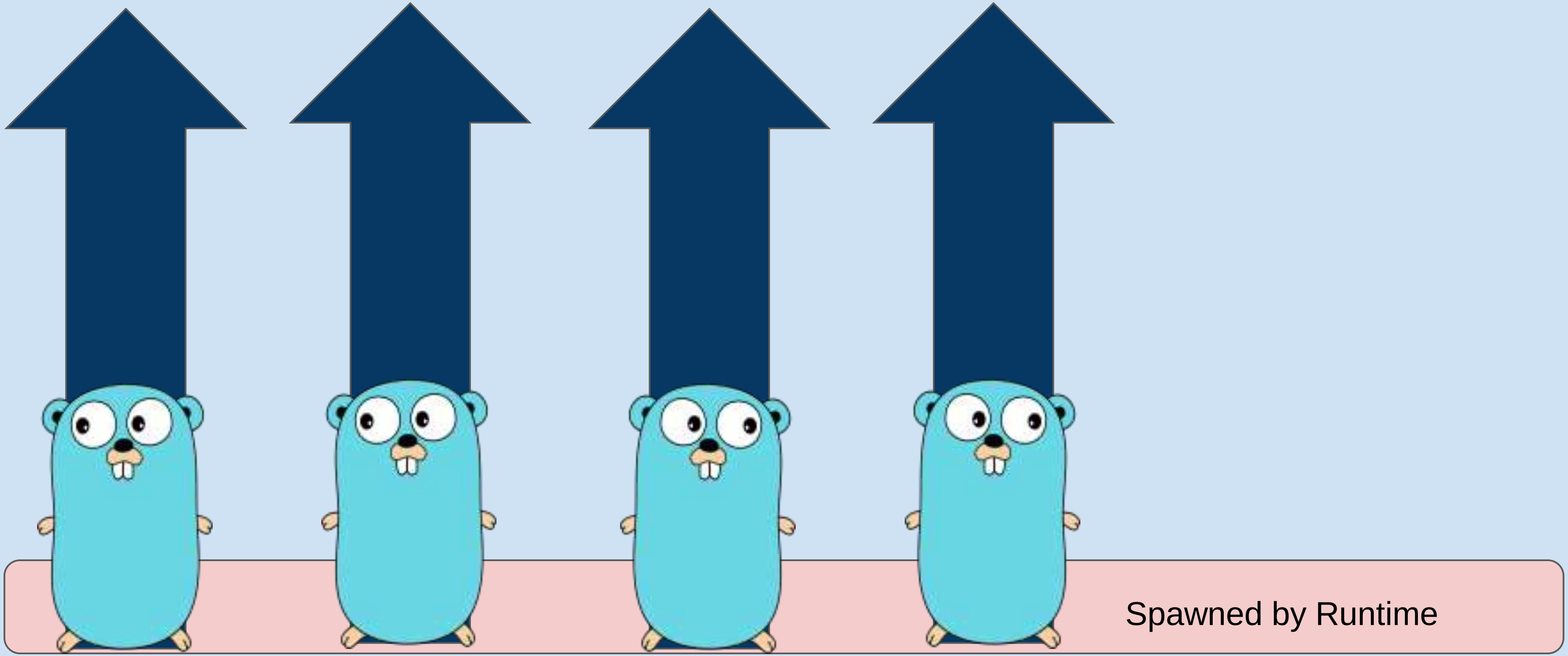
Goroutines



Goroutines

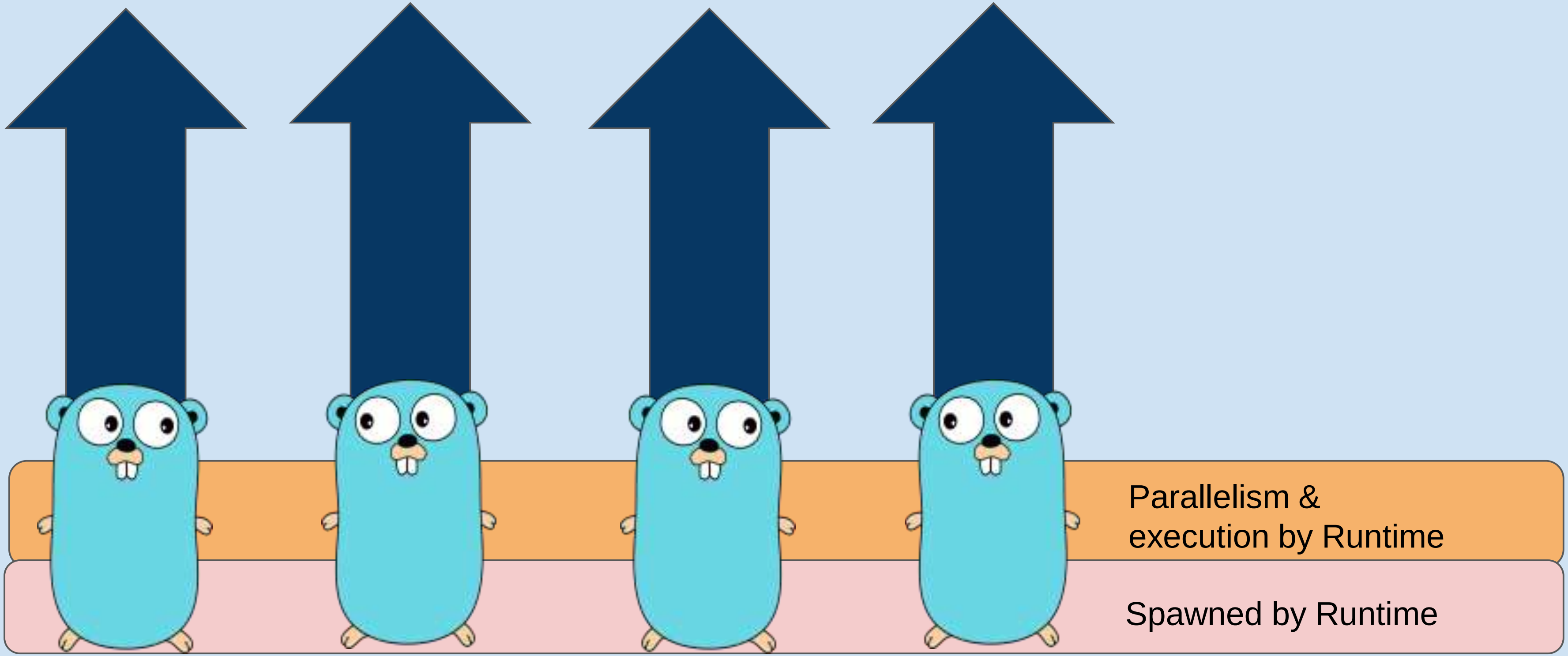


Goroutines

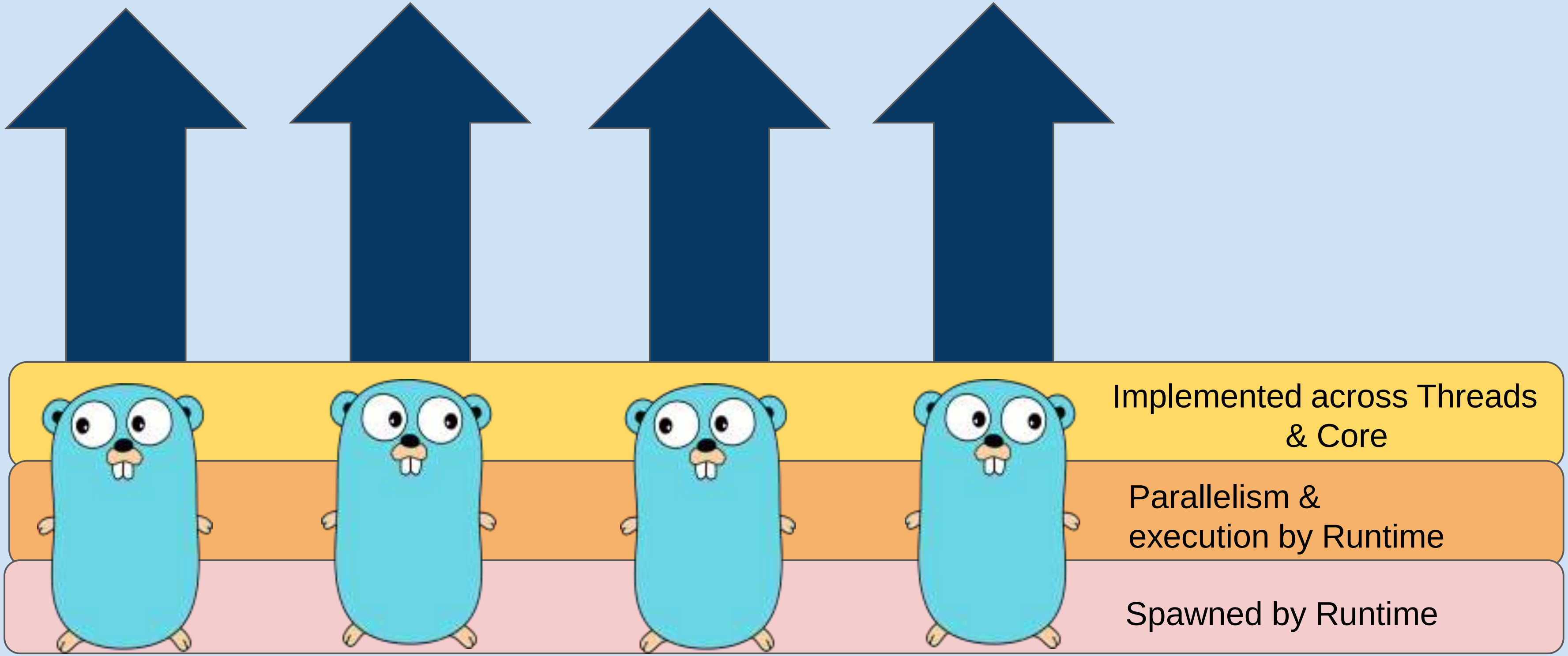


Spawned by Runtime

Goroutines



Goroutines

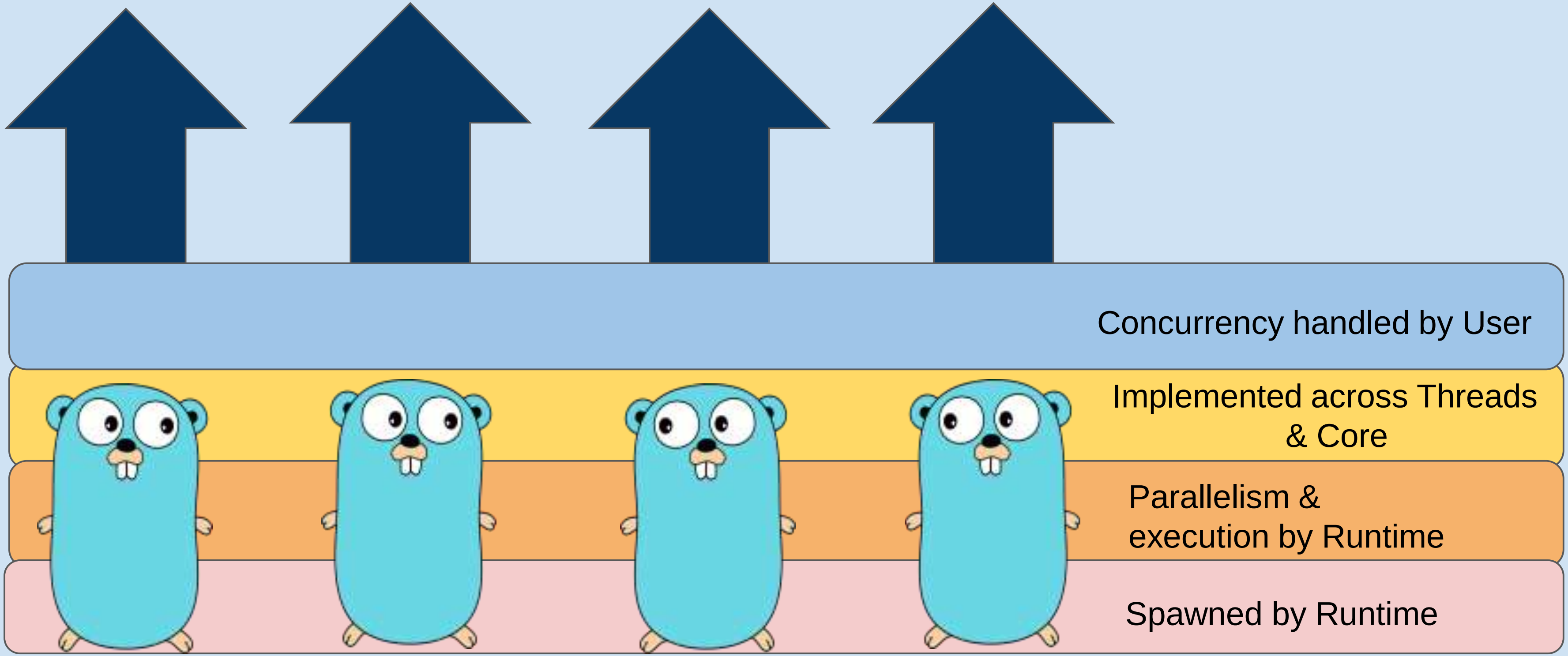


Implemented across Threads & Core

Parallelism & execution by Runtime

Spawned by Runtime

Goroutines



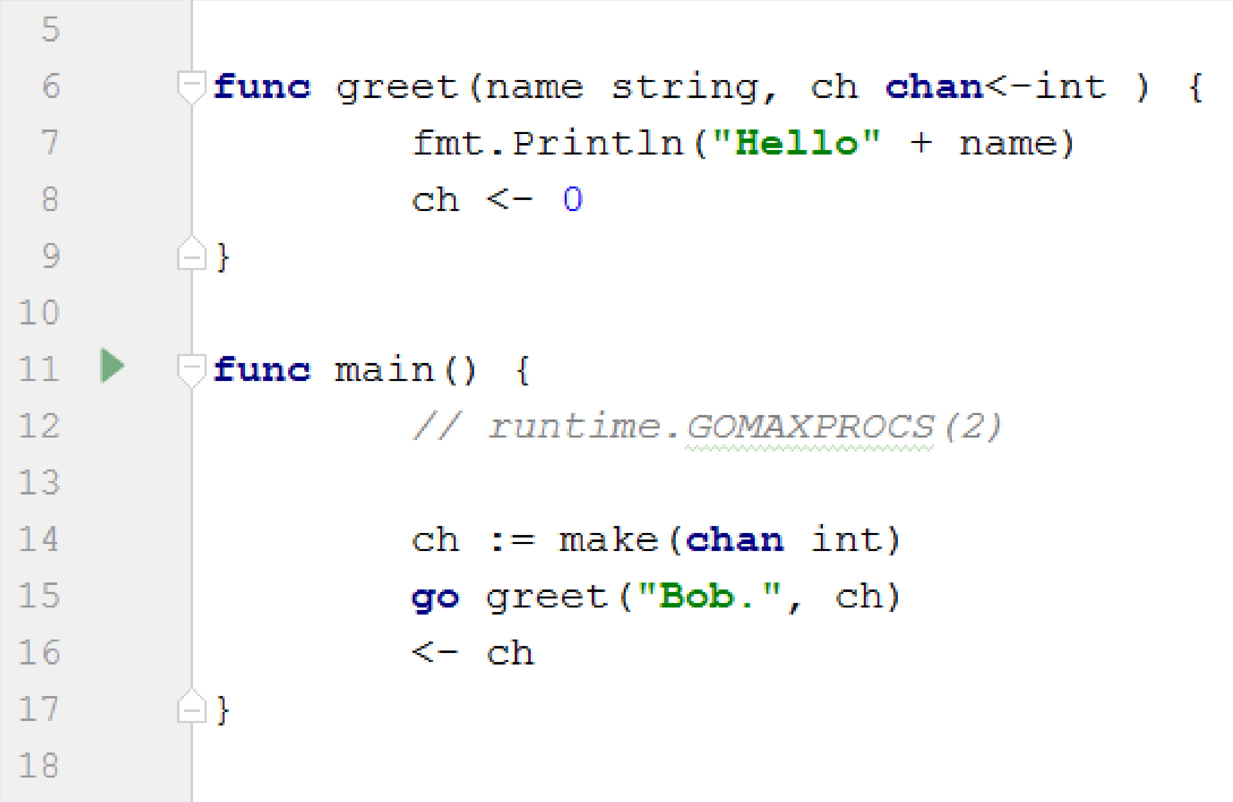
Concurrency handled by User

Implemented across Threads & Core

Parallelism & execution by Runtime

Spawned by Runtime

Goroutines



```go
func greet(name string, ch chan<-int ) {
        fmt.Println("Hello" + name)
        ch <- 0
}

func main() {
        // runtime.GOMAXPROCS(2)

        ch := make(chan int)
        go greet("Bob.", ch)
        <- ch
}
```

Pop Quiz

Goroutine Pop Quiz

Q: Since they are running across multiple threads, does failure in one goroutine halt the program?

Goroutine Pop Quiz

Q: Since they are running across multiple threads, does failure in one goroutine halt the program?

YES!

Goroutine Pop Quiz

Q: Will running CPU intensive tasks in goroutines block them?

Goroutine Pop Quiz

Q: Will running CPU intensive tasks in goroutines block them?

MAYBE?

I/O Bound == Concurrency

CPU Bound == Parallelism

Goroutines:
Coroutines + Green Threads
+ Much Much More!

- Twitter - @last_ent
- Github – github.com/last-ent