

Go Code

`go task1()`

`go task2()`

`go task3()`

`go task4()`

`go task5()`

`go build`

Go Runtime

Go Scheduler

Main G

 `task 2 go routine`

 `task 3 go routine`

 `task 5 go routine`

 `task 4 go routine`

 `task 1 go routine`

`./exec`

Executed

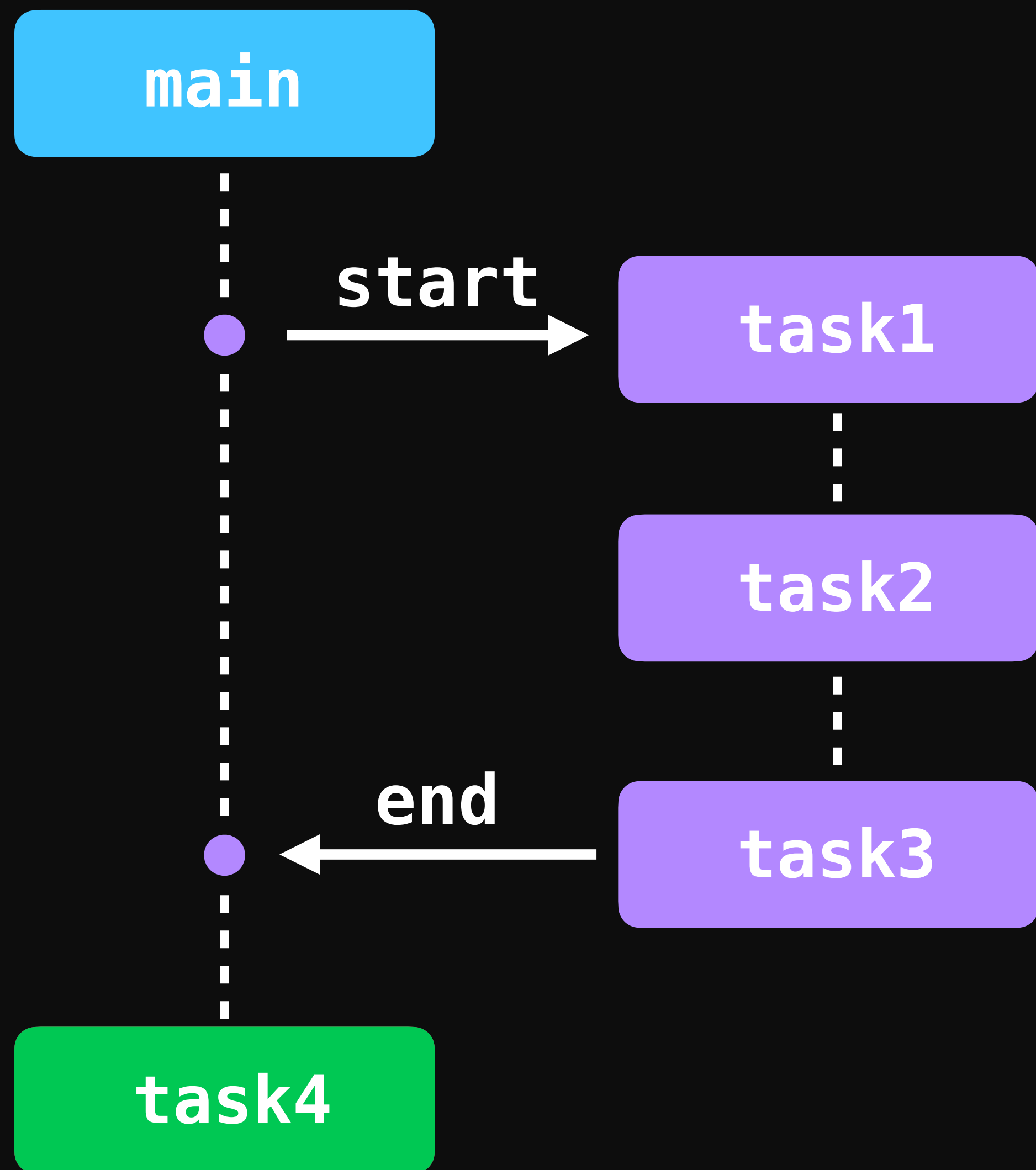
`task3`

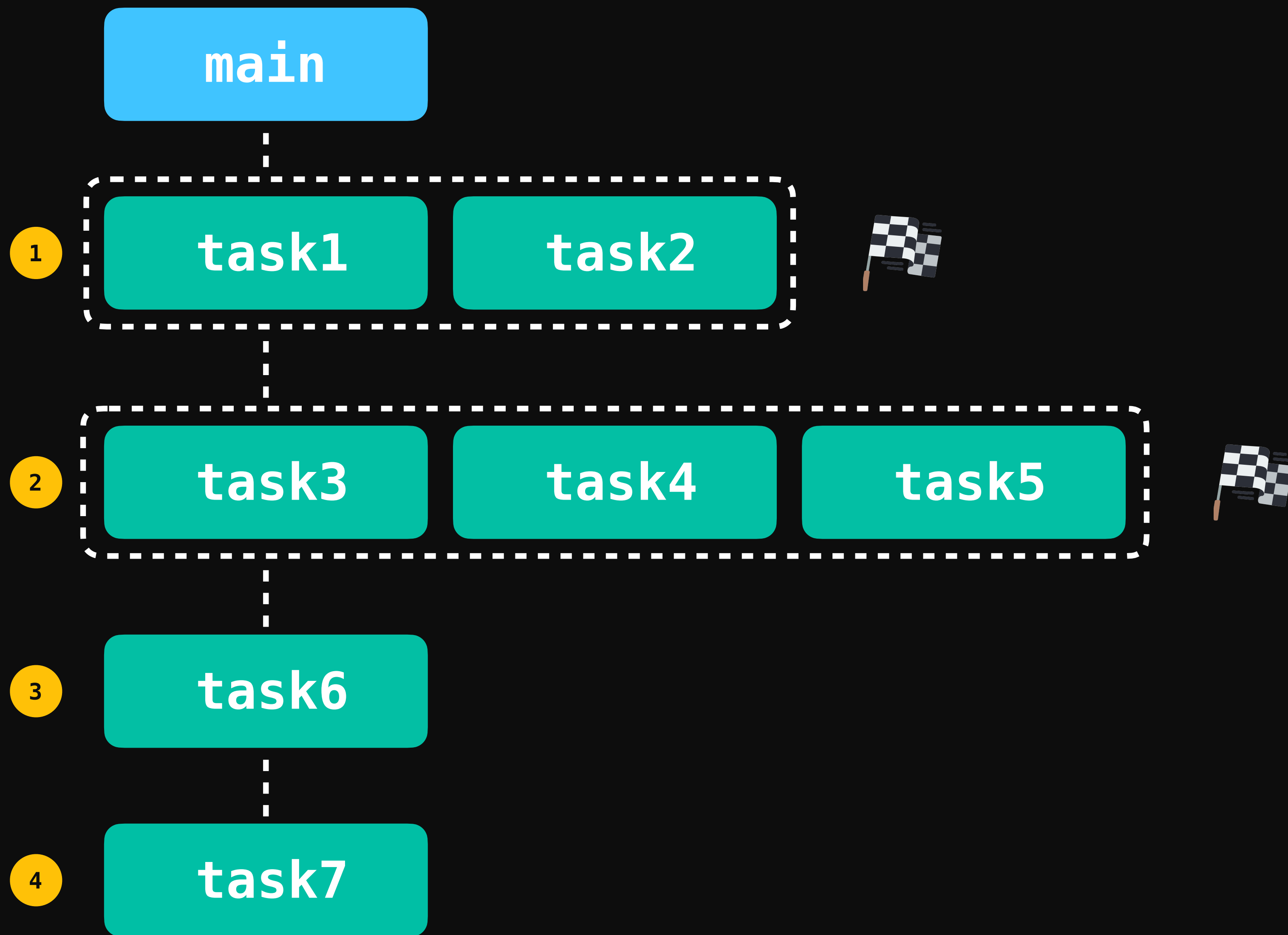
`task2`

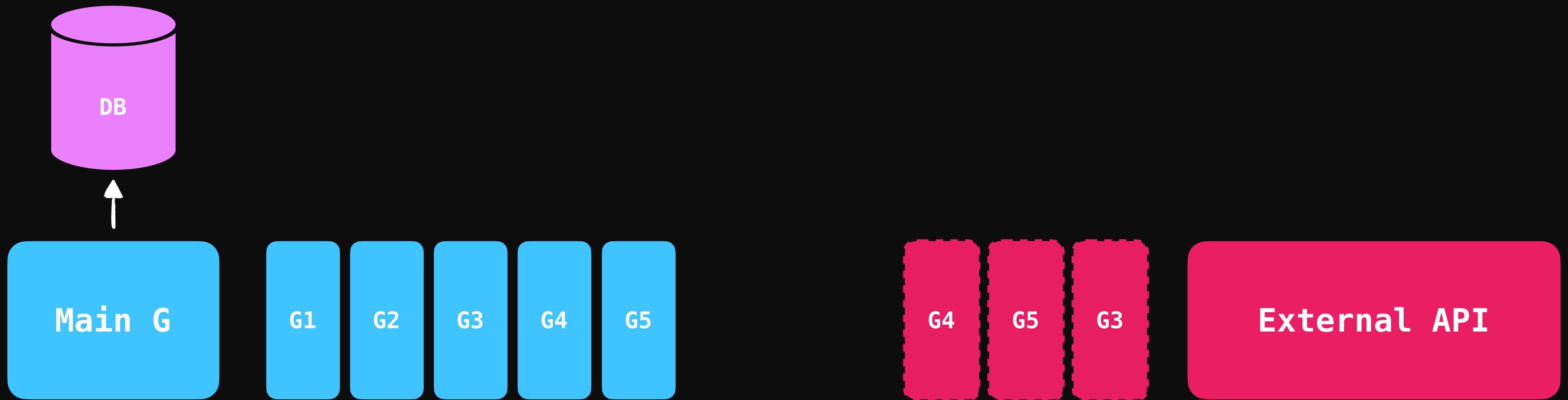
`task1`

`task4`

`task5`







WaitGroup

Mutex

RWMutex

Locker

Cond

Map

Pool

WaitGroup

count `int`

Add(`int`)

Increment `wg.counter`

Done()

Decrement `wg.counter`

Wait()

Exit when `wg.counter == 0`



Done() MUST be called as many times as **Add()**



Done() more than **Add()** => **panic**



Done() less than **Add()** => **deadlock**



Calling **Wait()** without **Add()** => **return immediately**



Prefer **Add(n)** vs **Add(1)** multiple times



WaitGroup MUST be passed by **reference**



Reusing **WaitGroup** before previous **Wait()** returns => **panic**