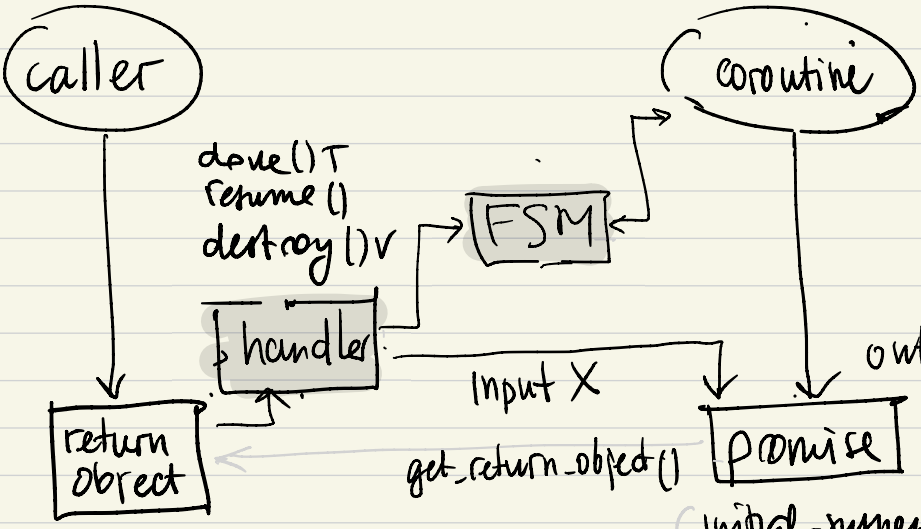


```
caller(){
auto ro = coroutine()

y = ro.next(x)
ro.handle_.promise().input_=x
ro.handle_.resume()  BLOCK
y = ro.handle_.promise().output_
```

```
return_object coroutine(){
  x = co_await
  promise.await_transform() → input_awaiter{}
    await_ready() → true
    await_suspend(h)  h_=h; return true.
    await_resume() → h_.promise().input_
        y = fn(x)
  co_yield y
  promise.yield_value(y) → output_awaiter{y}
    await_ready() → true
    await_suspend(h) h.promise().output_=y
                     return false
```