GO TEST

Presented by Michael Boudreau

AGENDA

- **Go's Built-in Facilities**
- Test Frameworks (if we have time)
- Out of Scope
  - Test Strategy & Practices

    (i.e. tdd vs bdd or when to mock, stub, spy, dummy, or fake)
  - How to test specific Go types

    (i.e. channels, interfaces, structs, slices)

TESTING IN GO

WHAT GO PROVIDES

- Simple CLI that just works
- Benchmarking
- Coverage analysis
- Race condition detection
- Example documentation

TESTING IN GO

```
$ go test
```

TESTING IN GO

THE BASICS

- Source File: `lottery.go`
- Test File: `lottery_test.go`
- Run Test: `go test`

TESTING IN GO

SOURCE FILE

```
// lottery.go
package lottery

func ProduceSixLotteryNumbers() []int {
   return []int{4,8,15,16,23,42}
}
```

TESTING IN GO

TEST FILE

```
// lottery_test.go
package lottery

import (
  "testing"
)

func TestXxx(t *testing.T) {
    //...
}
```

TESTING IN GO

```go
// lottery.go
package lottery

func ProduceSixLotteryNumbers() []int {
   return []int{4,8,15,16,23,42}
}
```

```go
// lottery_test.go
package lottery

func TestProduceSixLotteryNumbers(t *testing.T) {
    lottery := ProduceSixLotteryNumbers()

    if lottery == nil {
        t.Errorf("lottery should not be nil")
    } else if len(lottery) != 6 {
        t.Errorf("expected lottery to be size 6, but was %v", len(lottery))
    }
}
```

TESTING IN GO

RUN TEST

```
$ go test
$ go test -v
```

BENCHMARKING IN GO

```
$ go test -bench .
```

BENCHMARKING IN GO

TEST FILE

```go
// lottery_test.go
package lottery

import (
  "testing"
)

func BenchmarkXxx(b *testing.B) {
    for i := 0; i < b.N; i++ {
        //
    }
}
```

BENCHMARKING IN GO

TEST AND BENCHMARK FUNCTIONS

```
func TestProduceSixLotteryNumbers(t *testing.T) {
    lottery := ProduceSixLotteryNumbers()

    if lottery == nil {
        t.Errorf("lottery should not be nil")
    } else if len(lottery) != 6 {
        t.Errorf("expected lottery to be size 6, but was %v", len(lottery))
    }
}

func BenchmarkProduceSixLotteryNumbers(b *testing.B) {
    for i := 0; i < b.N; i++ {
        ProduceSixLotteryNumbers()
    }
}
```

BENCHMARK EXAMPLE

COVERAGE ANALYSIS IN GO

SIMPLE AND INTUITIVE

```
$ go test -cover
```

COVERAGE ANALYSIS IN GO

OUTPUT VIEWING OPTIONS

```
$ go test -coverprofile=cover.out
 $ go tool cover -func=cover.out
 $ go tool cover -html=cover.out
```

COVERAGE EXAMPLE

RACE CONDITION DETECTION IN GO

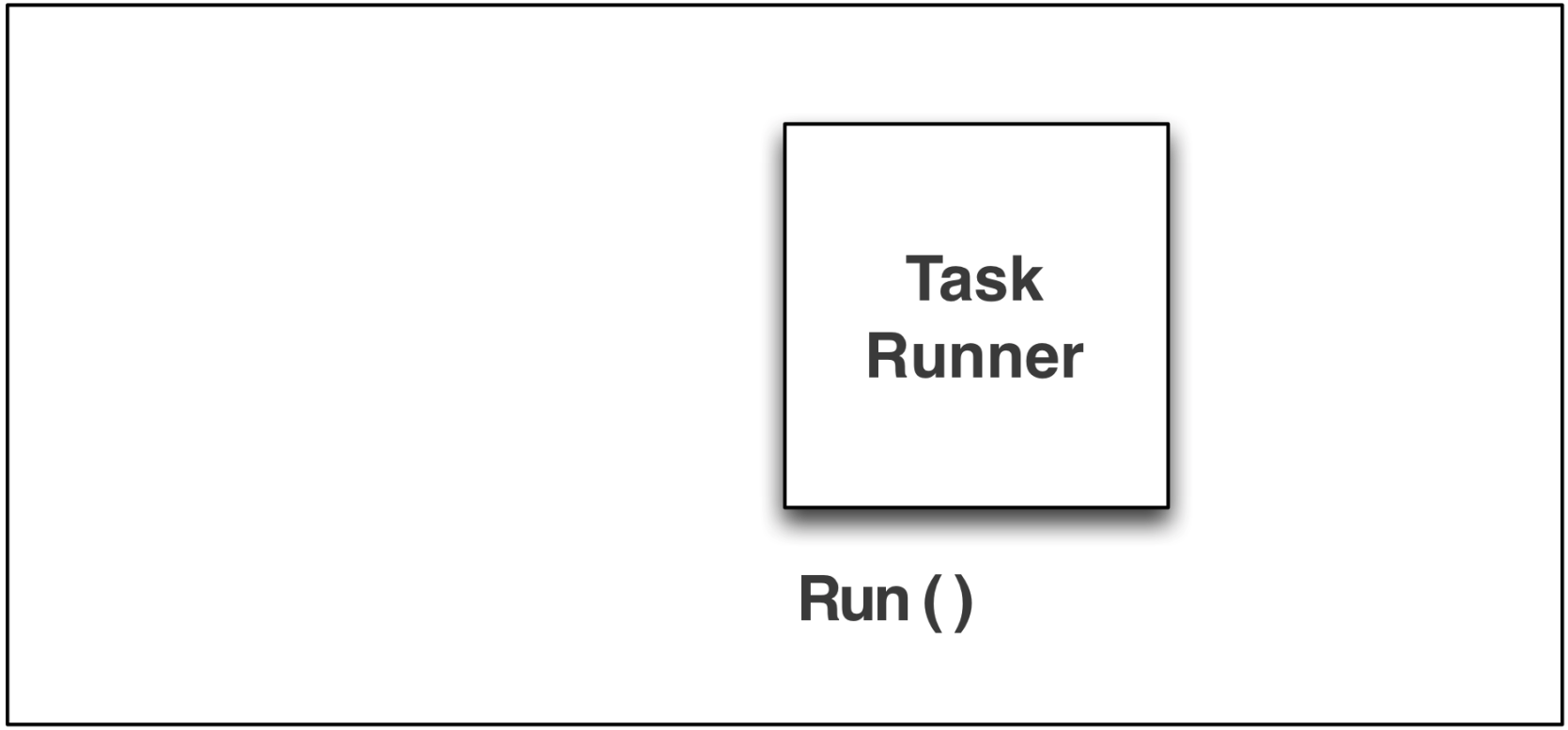
```
$ go test -race
```

RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER

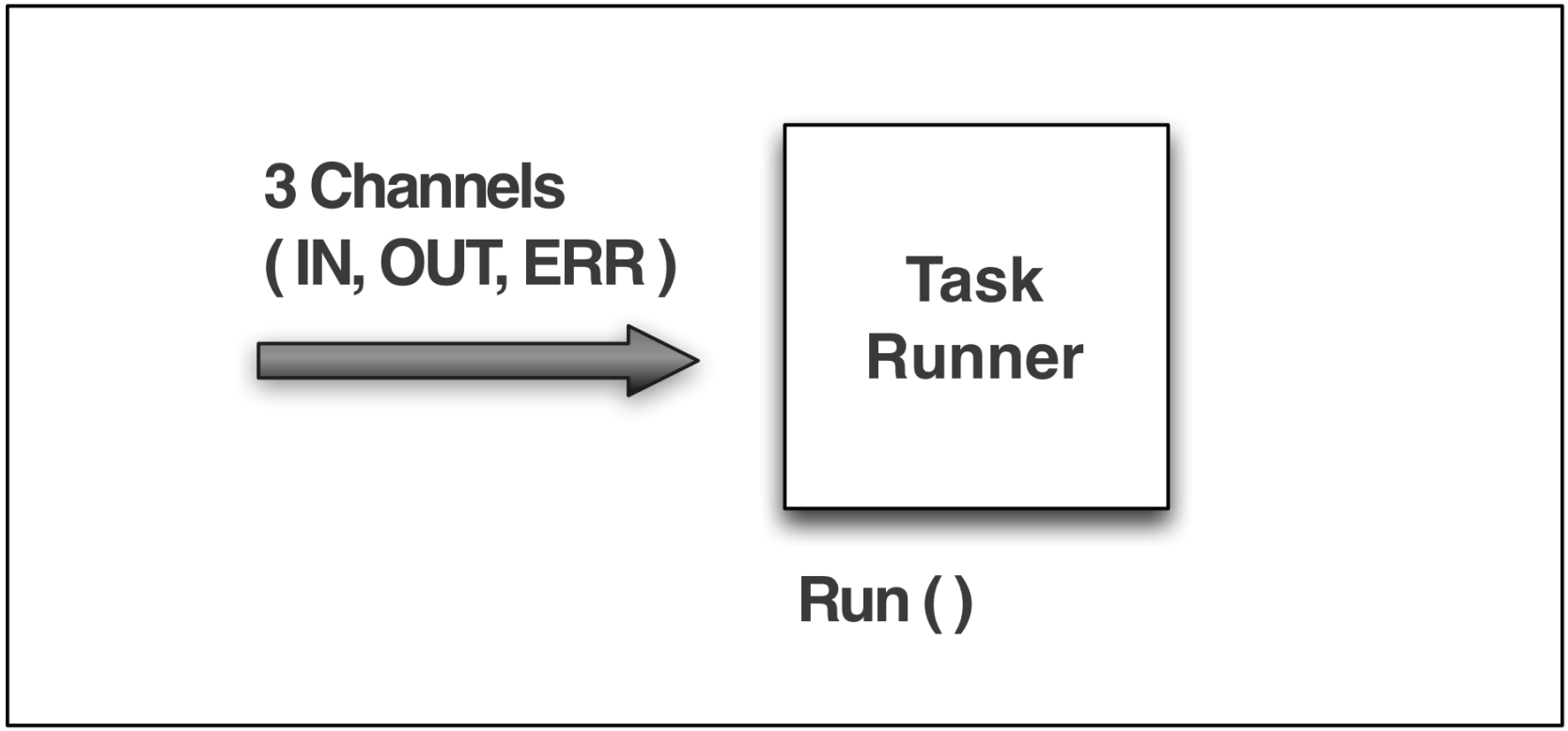
RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER

```
type TaskData struct {
  In    <-chan interface{}
  Out   chan<- interface{}
  Error chan<- error
}

type TaskRunner interface {
  Run(data *TaskData)
}

func RunTask(data *TaskData, runner TaskRunner) {
  go func() {
    defer close(data.Out)
    runner.Run(data)
  }()
}
```

RACE CONDITION DETECTION IN GO

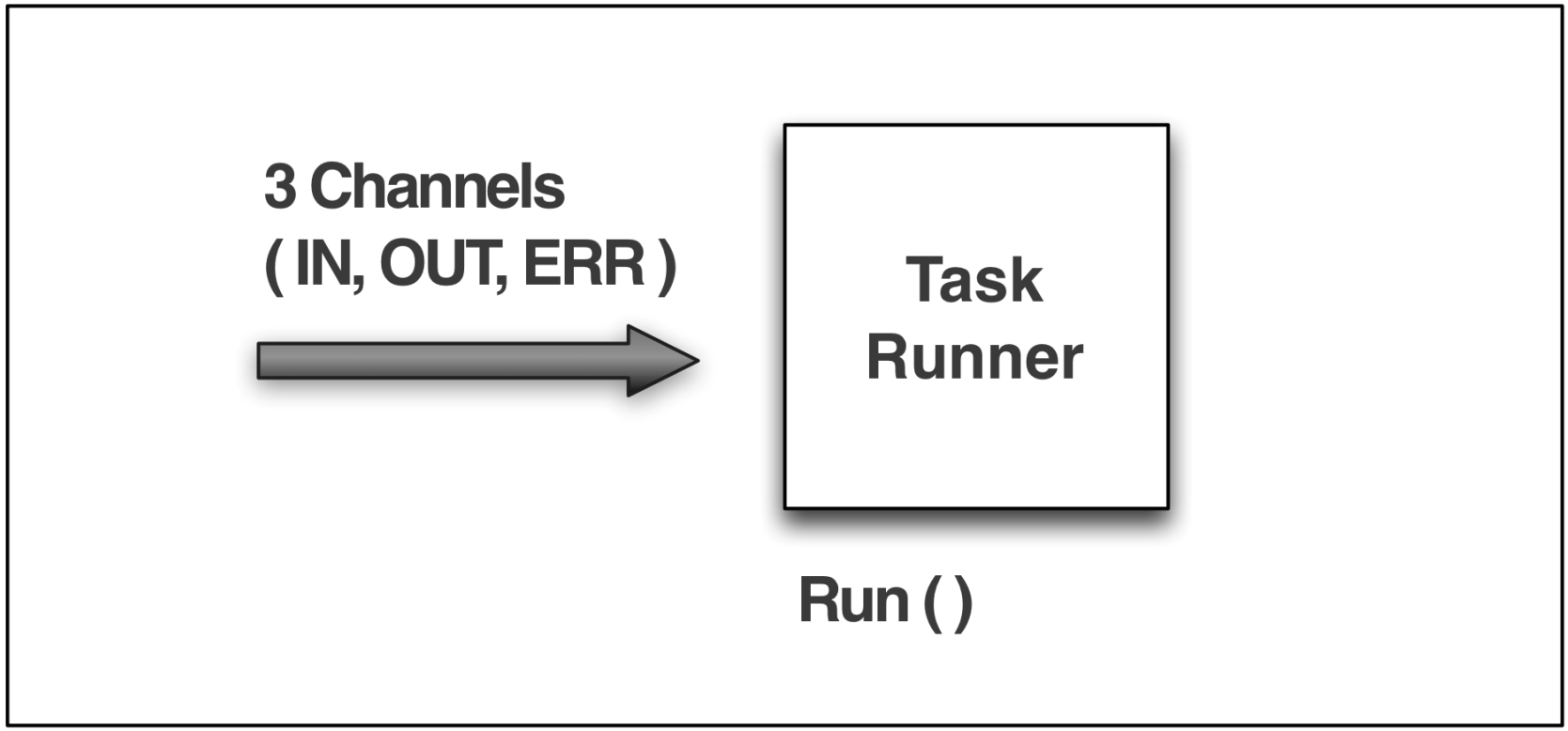
CONTEXT: MY TASK RUNNER

```
in := make(chan interface{})
out := make(chan interface{})
err := make(chan error)
taskData := &TaskData{
  In:    in,
  Out:   out,
  Error: err,
}

RunTask(taskData, &FilterString{Filter: "HELLO"})
go func() {
    in <- "TESTING A"
    in <- "123"
    in <- "HELLO"
    in <- "456"
    in <- "TESTING B"
    close(in)
}()
```

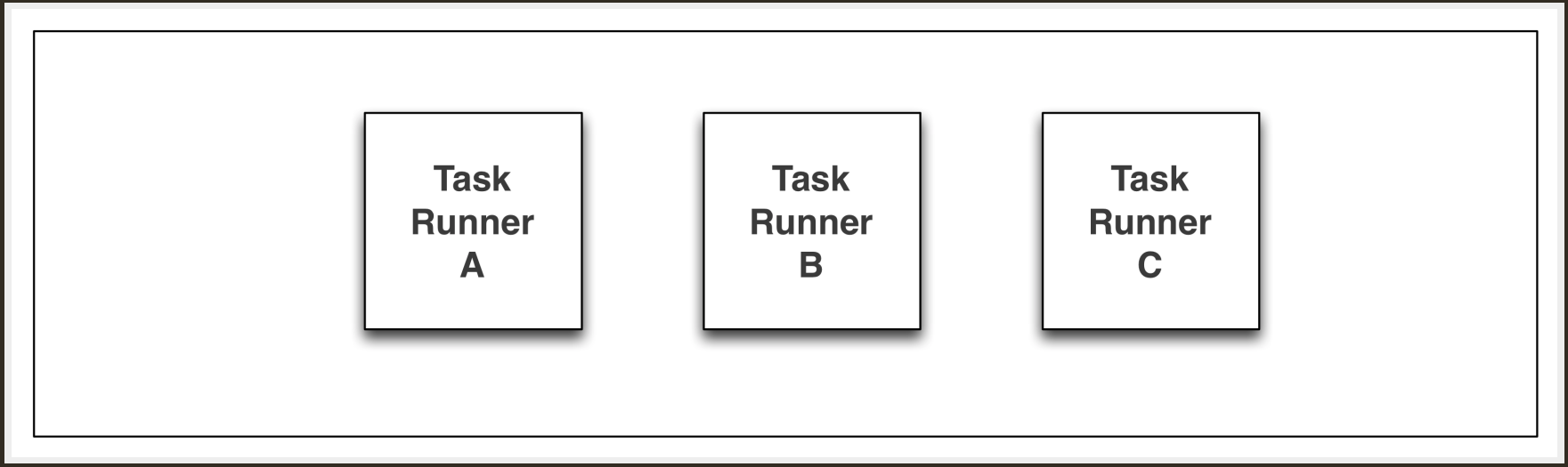
RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



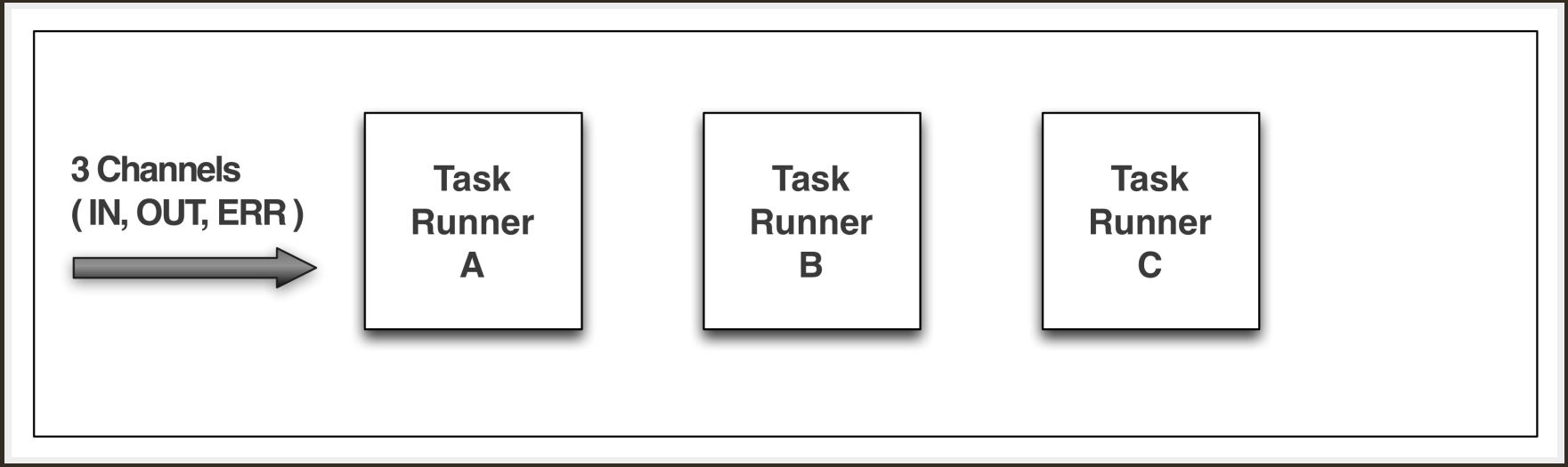
RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



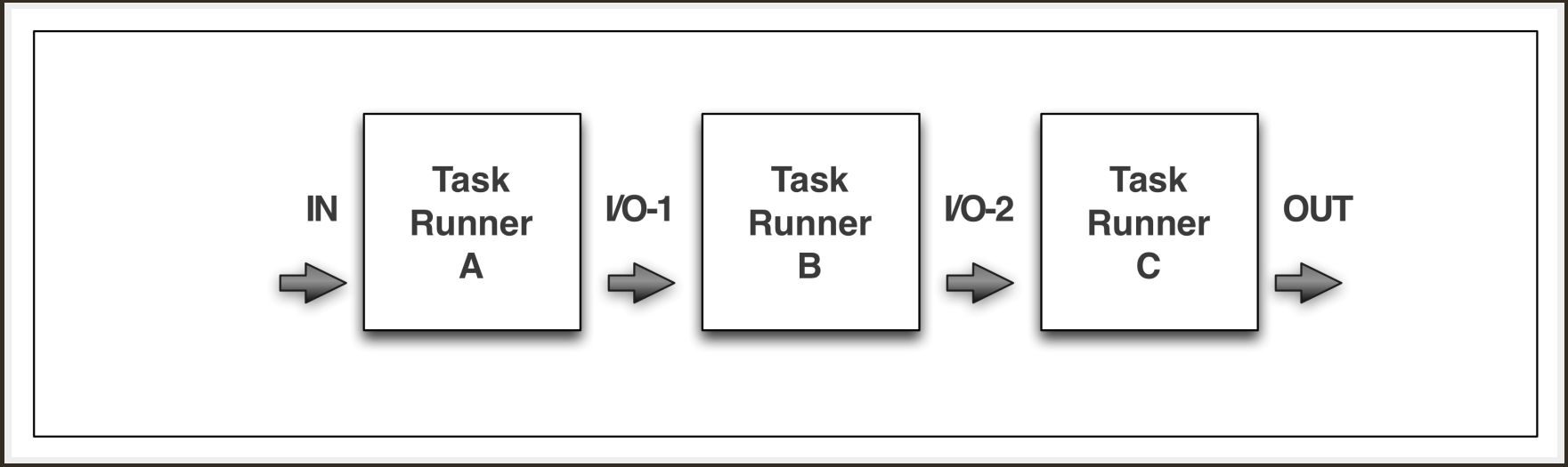
RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER



RACE CONDITION DETECTION IN GO

CONTEXT: MY TASK RUNNER

```
func ChainTasks(data *TaskData, runners ...TaskRunner) {
  for i, runner := range runners {
    if i < len(runners)-1 {
      ch := make(chan interface{})
      RunTask(&TaskData{In: data.In, Out: ch, Error: data.Error}, runner)
      data.In = ch
    } else {
      RunTask(&TaskData{In: data.In, Out: data.Out, Error: data.Error}, runne
    }
  }
}
```

RACE DETECTION EXAMPLE

RACE OUTPUT

MAIN SECTIONS

```
==================
WARNING: DATA RACE
Write by goroutine 5:
  // ...

Previous read by goroutine 7:
  // ...

Goroutine 5 (running) created at:
  // ...

Goroutine 7 (running) created at:
  // ...
==================
.........................................PASS
Found 1 data race(s)
```

RACE OUTPUT

WRITE

```
Write by goroutine 5:
  runtime.closechan()
  .../code/taskrunner.func 001()
      .../code/taskrunner/taskrunner.go:22 +0xba
```

READ

```
Previous read by goroutine 7:
  runtime.chansend()
  .../code/taskrunner.adaptStringChannelToInterfaceChannel()
      .../code/taskrunner/taskrunner_string.go:47 +0xbd
```

RACE OUTPUT

WRITE BY GOROUTINE 5: RUNTIME.CLOSECHAN()

```
  .../code/taskrunner.func 001()
      /Users/.../code/taskrunner/taskrunner.go:22 +0xba
```

```
func RunTask(data *TaskData, runner TaskRunner) {
    go func() {
        defer close(data.Out)
        runner.Run(data)
    }() // line 22
}
```

PREVIOUS READ BY GOROUTINE 7: RUNTIME.CHANSEND()

```
  .../code/taskrunner.adaptStringChannelToInterfaceChannel()
      /Users/.../code/taskrunner/taskrunner_string.go:47 +0xbd
```

```
func adaptStringChannelToInterfaceChannel(in <-chan string, out chan<- interf
    defer close(out)

    for someString := range in {
        out <- someString // line 47
    }
}
```

RACE DETECTION EXAMPLE CONTINUED

DOCUMENTATION EXAMPLES IN GO

EXAMPLES IN GO

TEST FILE

```
package helloworld

import (
    "fmt"
)

func Example() {
    result := DoSomething()
    fmt.Println(result)

    // Output: Hello
}
```

EXAMPLES IN GO

FORMAT OF FUNCTION NAMES

- Package Level Example

```
func Example() {}
```

- Example for Function

```
func ExampleFuncName() {}
```

- Example for Type

```
func ExampleTypeName() {}
```

- Example for Method on Type

```
func ExampleTypeName_MethodName() {}
```

- Example for Variant of Function (applies to all)

```
func ExampleFuncName_variation() {}
```

EXAMPLES IN GO

```
godoc -http=:8888
```

TEST FRAMEWORKS & LIBRARIES IN GO

TEST FRAMEWORKS & LIBRARIES IN GO

FRAMEWORKS & LIBRARIES

- Ginkgo
- Goblin
- Gospecify
- Gocheck
- Goconvey
- oglematchers
- Gomega
- Gocov
- Gomock

TEST FRAMEWORKS & LIBRARIES IN GO

PROS & CONS

- Expressive
- Additional syntax and mini-language to learn
- Devs familiar with syntax style
- Potential complicated setup
- Setup & Teardown per suite or per test
- Additional dependencies
- Usually compatible with Go's test facilities

TEST FRAMEWORKS & LIBRARIES IN GO

GINKGO TEST FRAMEWORK & GOMEGA MATCHERS

```
func TestExamplePackage(t *testing.T) {
    RegisterFailHandler(Fail)
    RunSpecs(t, "Example Package Suite")
}
```

```
var _ = Describe("some description.... ", func() {
    Context("Some context....", func() {
        It("Should work with table-driven tests", func() {
            for _, testcase := range testcases {
                actual := MyFunction(testcase.input)

                Expect(actual).ToNot(BeNil())
                Expect(actual).To(Equal(testcase.expected))
            }
        })
        It("Should be nil with the text inconceivable", func() {
            badTestCase := "inconceivable"
            actual := MyFunction(badTestCase)
            Expect(actual).To(BeNil())
        })
    })
})
```

TEST FRAMEWORKS & LIBRARIES IN GO

GOCONVEY TEST FRAMEWORK & OGLEMATCHERS MATCHERS

```
func TestSpec(t *testing.T) {
    Convey("Given some integer with a starting value", t, func() {
        x := 1

        Convey("When the integer is incremented", func() {
            x++

            Convey("The value should be greater by one", func() {
                So(x, ShouldEqual, 2)
            })
        })
    })
}
```

TEST FRAMEWORKS & LIBRARIES IN GO

MY OPINIONS

- Enhances Readability; hence, maintainability
- Adds Value
- Utilizes Go's Testing Facilities

THANK YOU