Go program anatomy

PROGRAM LAYOUT

main.go

```
package main
```

* REQUIRED

```
import (
  ...
)
```

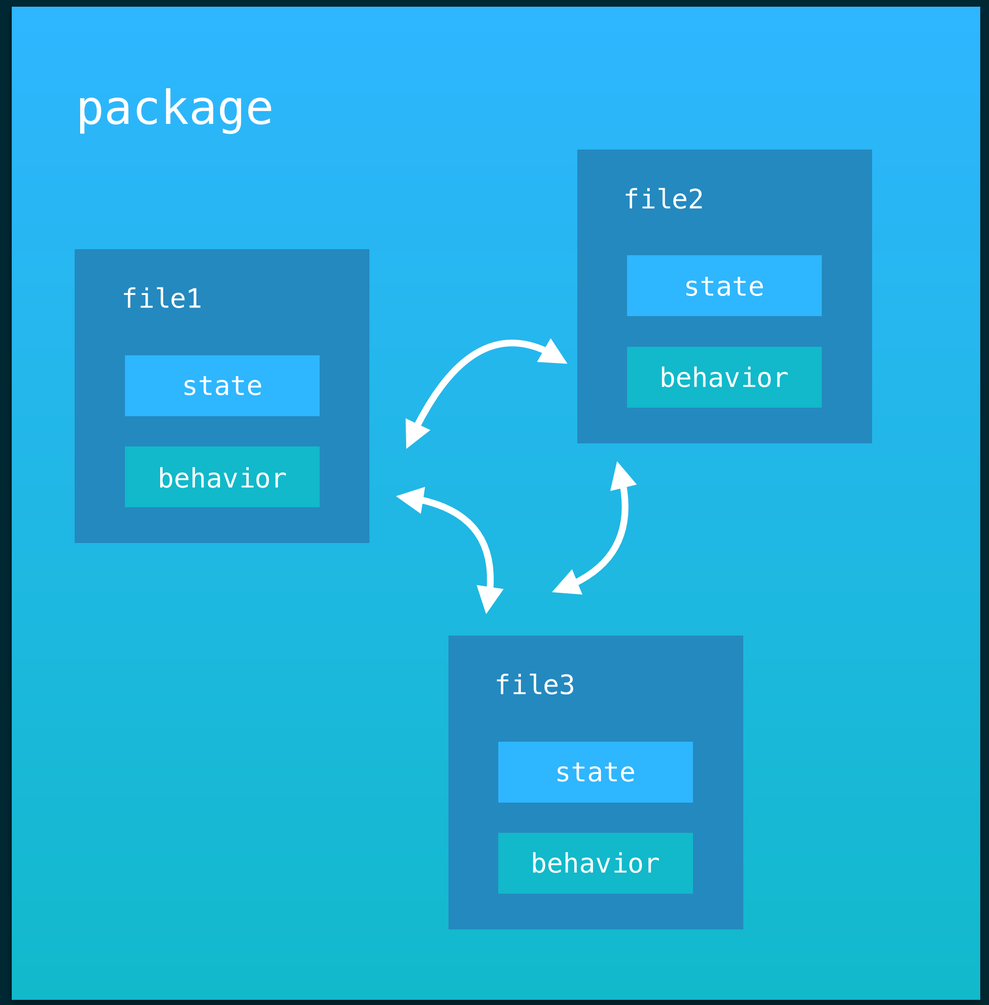
```
const ...
```

```
var ...
```

```
func init() {
  // initialize
}
```

```
func main() {
  // execute
}
```

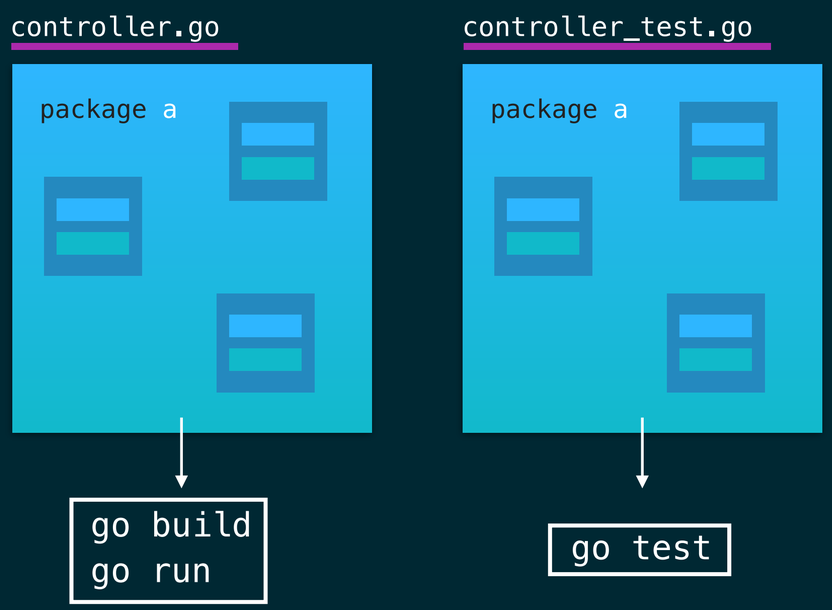
PACKAGE ANATOMY



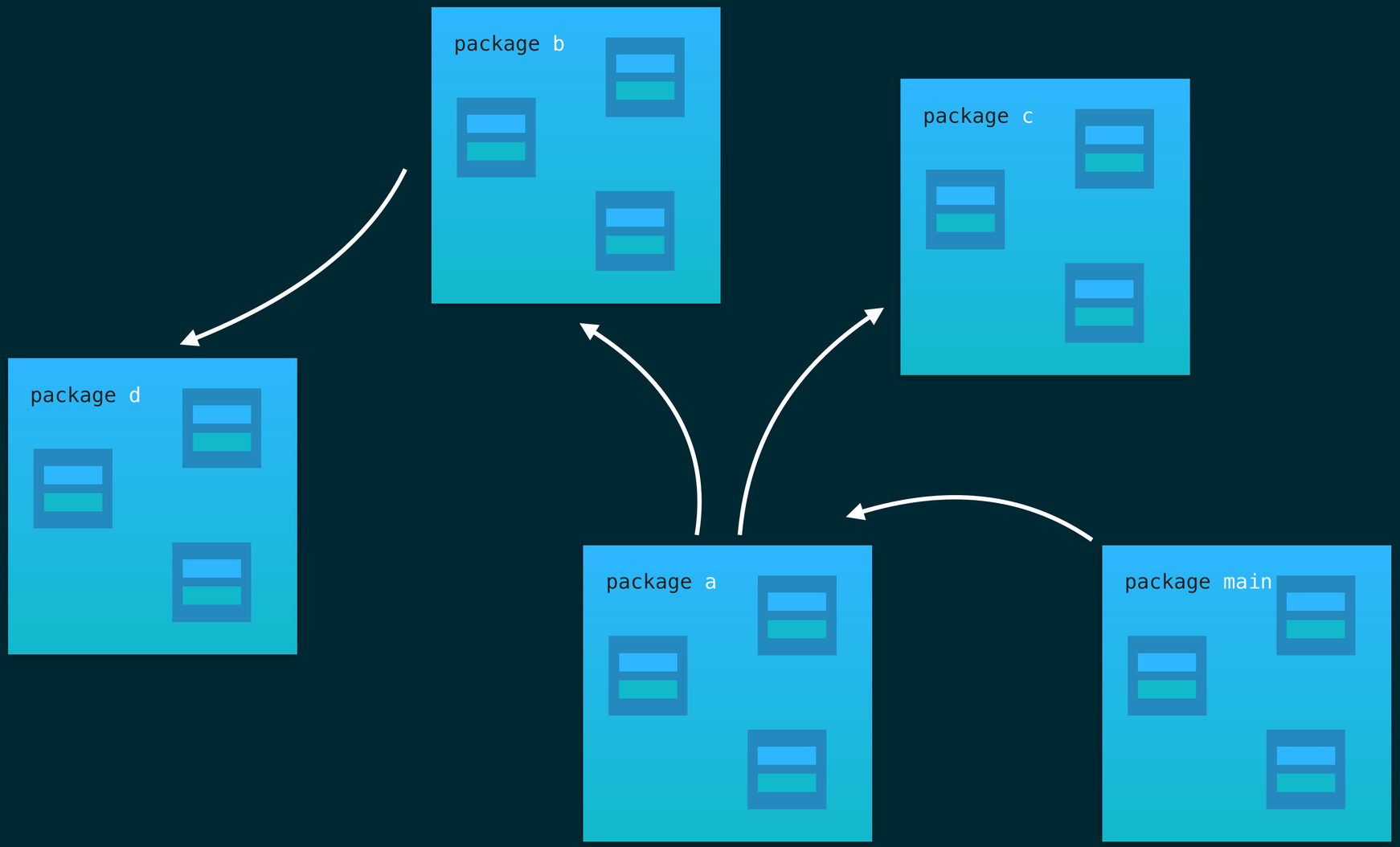
state

behavior

FILE TYPES

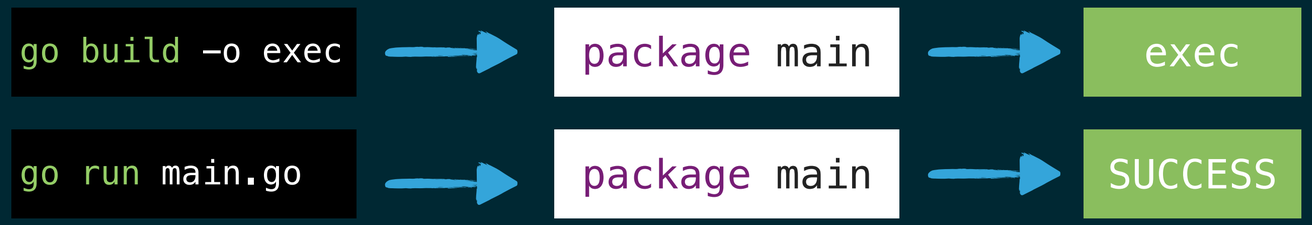

controller.go
package a
go build
go run
controller_test.go
package a
go test

GO PROGRAM

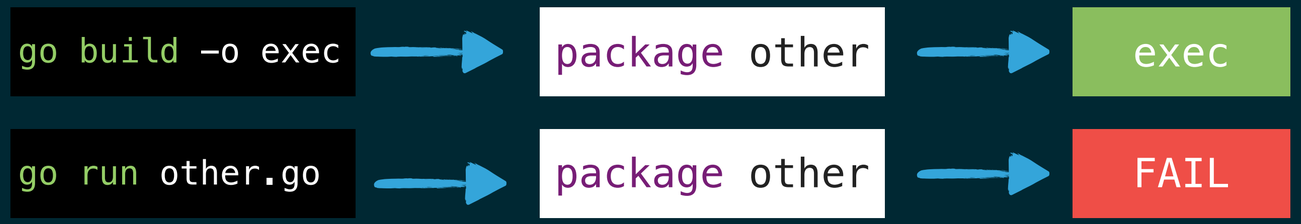


PACKAGE TYPES

executable (main)

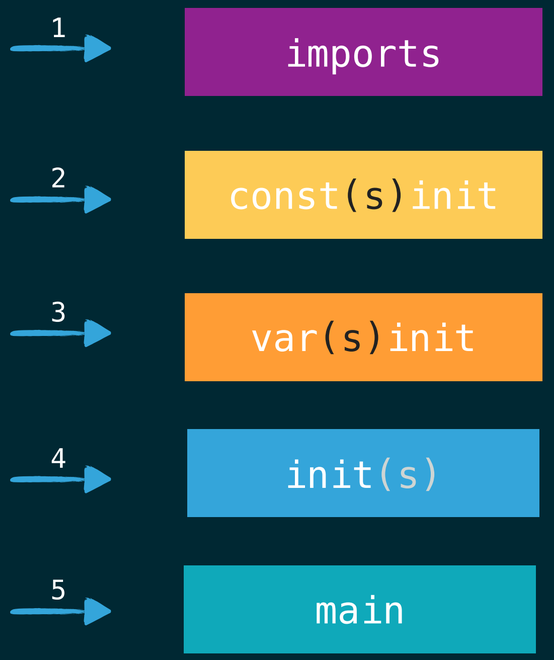


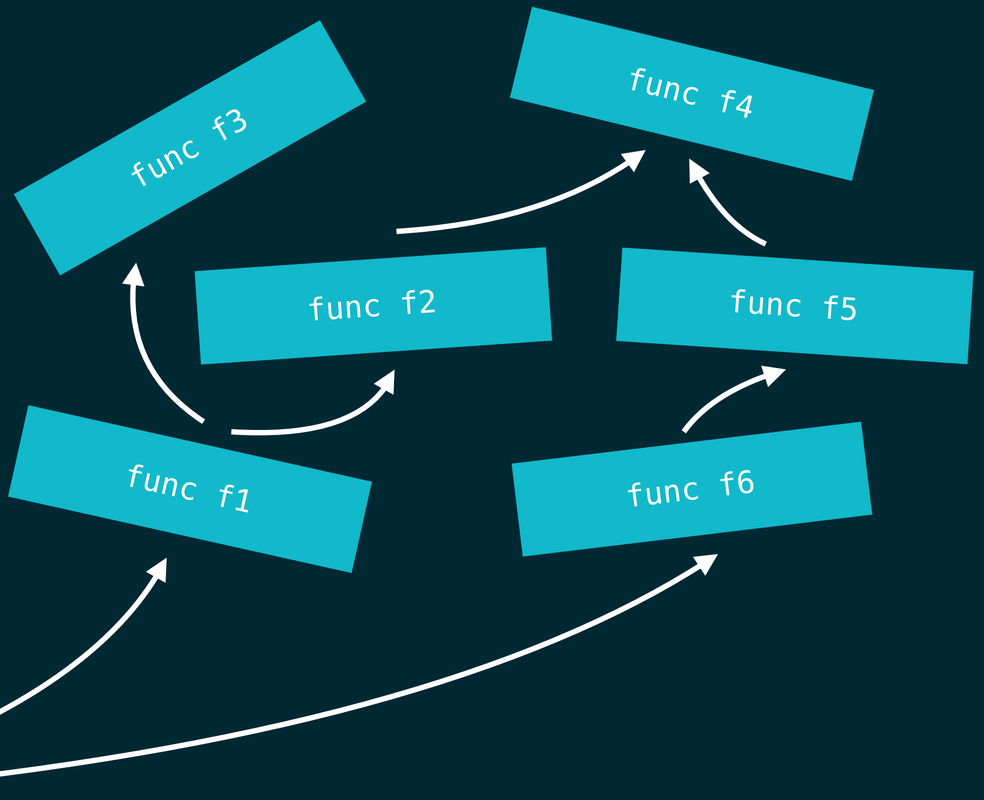
un executable (non main)



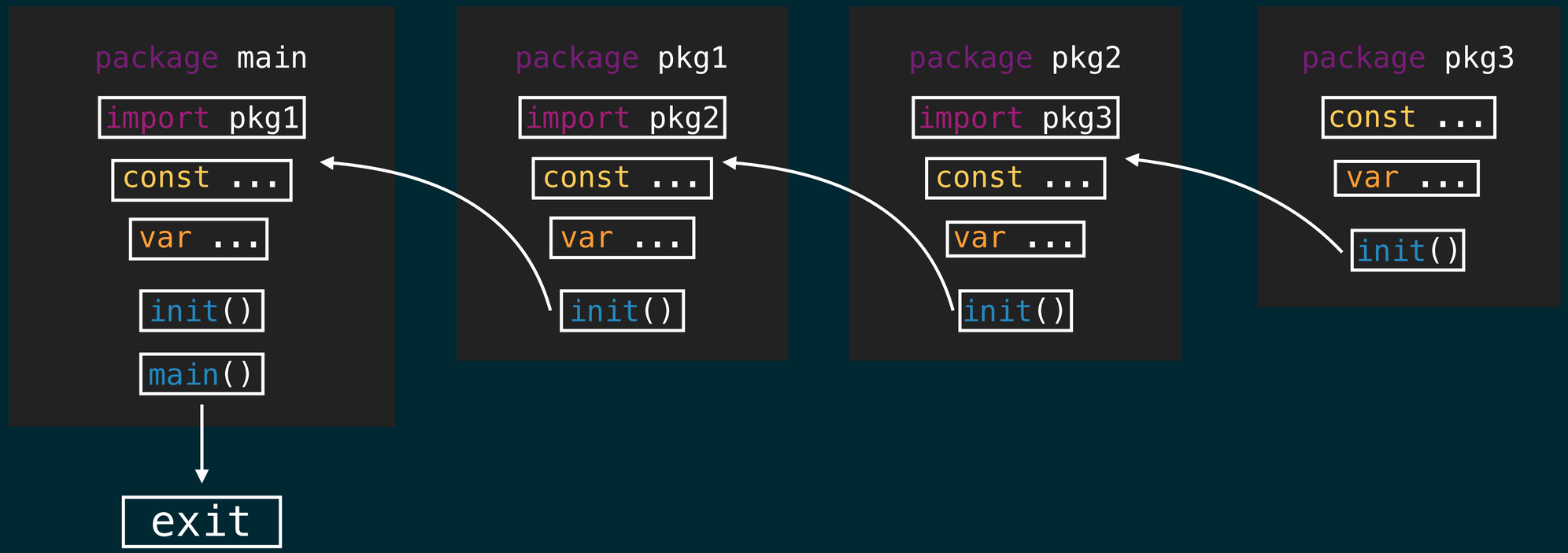
PROGRAM LIFECYCLE

```
package main

import (
  ...
)

const ...
var ...

func init() {
  // initialize
}

func main() {
  // execute
}
```





FLOW EXAMPLE



GO BUILD / TAGS

2 Explicit $GOOS & $GOARCH

1 Default $GOOS & $GOARCH

3 File names or tags

```
env GOOS=linux
GOARCH=amd64
```

↓

```
$GOOS $GOARCH
```

↓

```
go build
```

↓

```
platform-binary
```

SUPPORTED ARCH

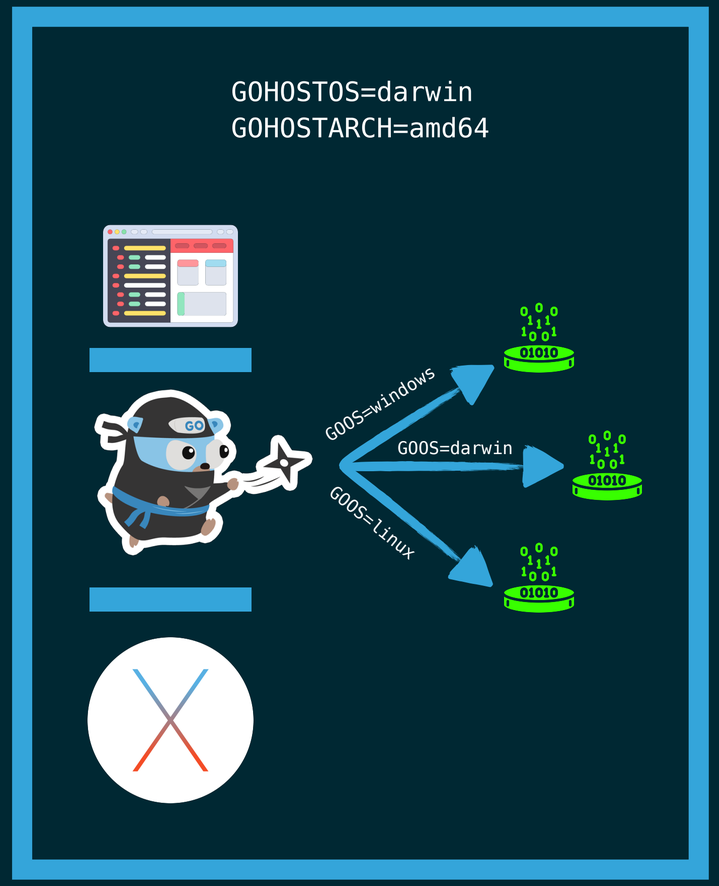
File name suffix	File tags
FILE_GOOS.go	// +build os1[,arch1] [os2[,arch2]]
FILE_GOARCH.go	// +build !linux,!darwin !cgo
FILE_GOOS_GOARCH.go	// +build ignore
	// +build custom-tag

comma AND

space OR

BUILD/COMPILE TIME VS RUNTIME

BUILD TIME



RUN TIME