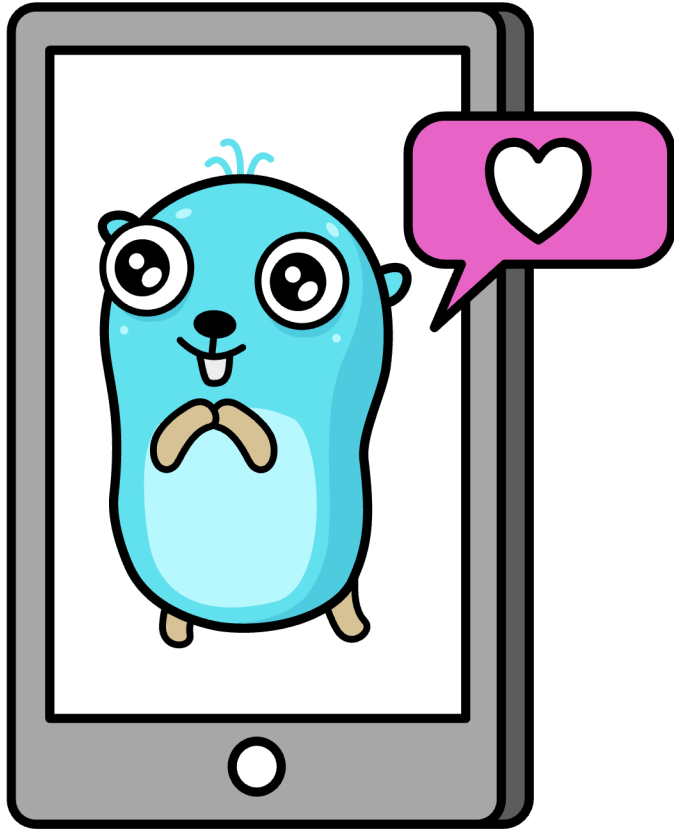


Implement FTP with

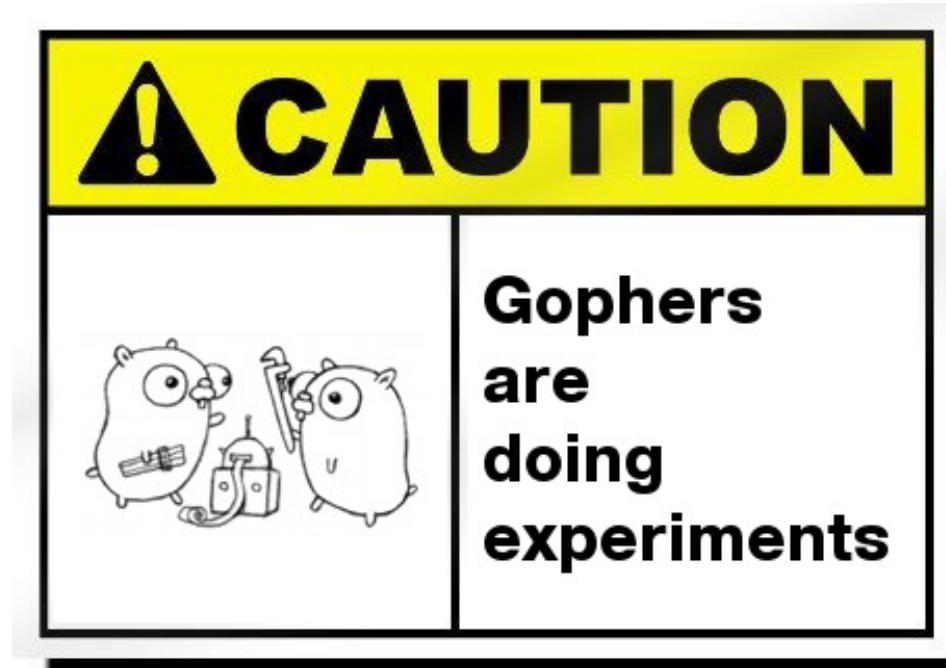
刘乔升 / 佘家瑞

 <https://github.com/qsliu2017/FxxkTheProject>

Gomobile: from Golang to AAR

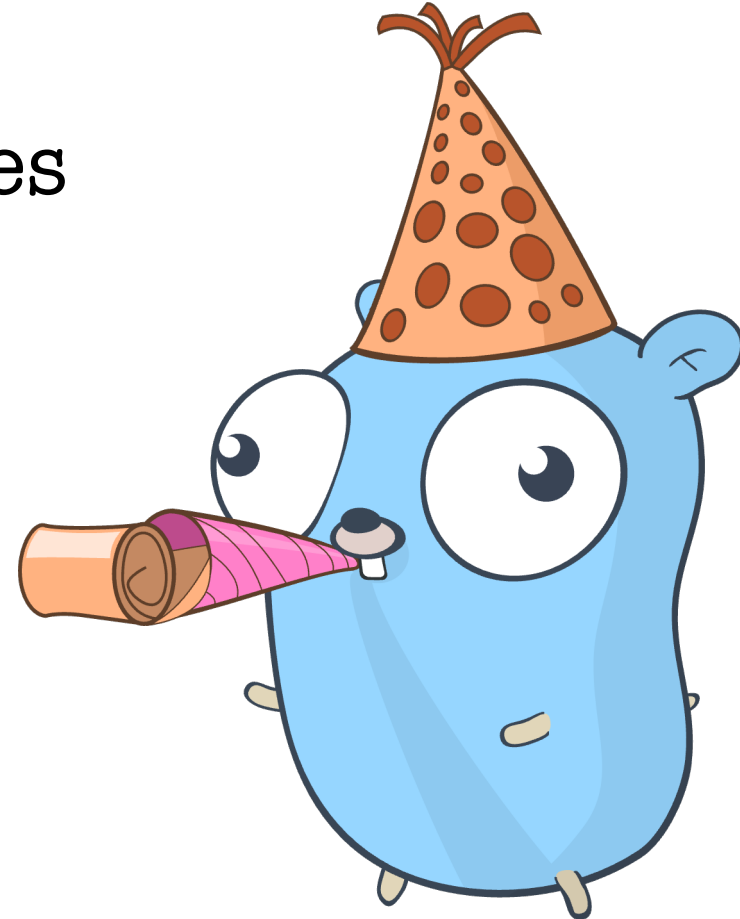
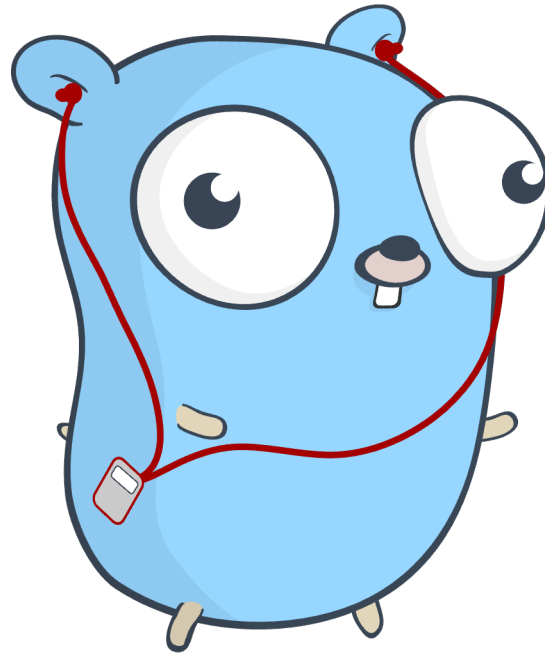


go mobile()



3 Reasons to Use Golang

- Build-in Goroutine Support
- Well-designed IO and Network Packages
- Gophers are CUTE!



Build-in Goroutine Support

```
// To run a function parallel  
go handleClient()
```

```
// Use channel for communicating  
msgChannel := make(chan string)
```

```
msg := <- msgChannel
```

```
msgChannel <- msg
```



go Friday()

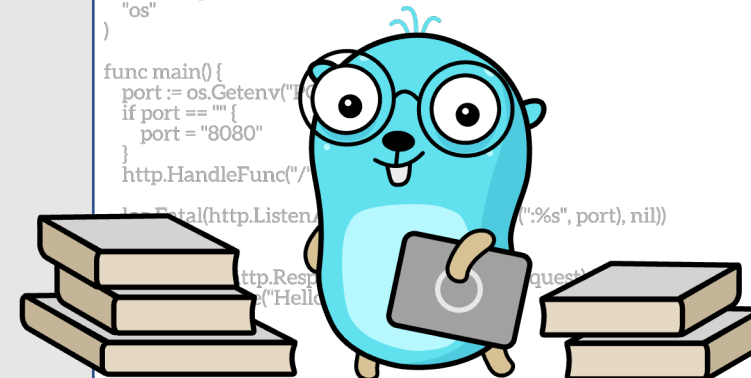


go space()

```
// This server can run on App Engine.  
package main
```

```
import (  
    "fmt"  
    "log"  
    "net/http"  
    "os"  
)
```

```
func main() {  
    port := os.Getenv("PORT")  
    if port == "" {  
        port = "8080"  
    }  
    http.HandleFunc("/",  
        func(w http.ResponseWriter, r *http.Request) {  
            fmt.Fatal(w, http.ListenAndServe(port, nil))  
        })  
}
```

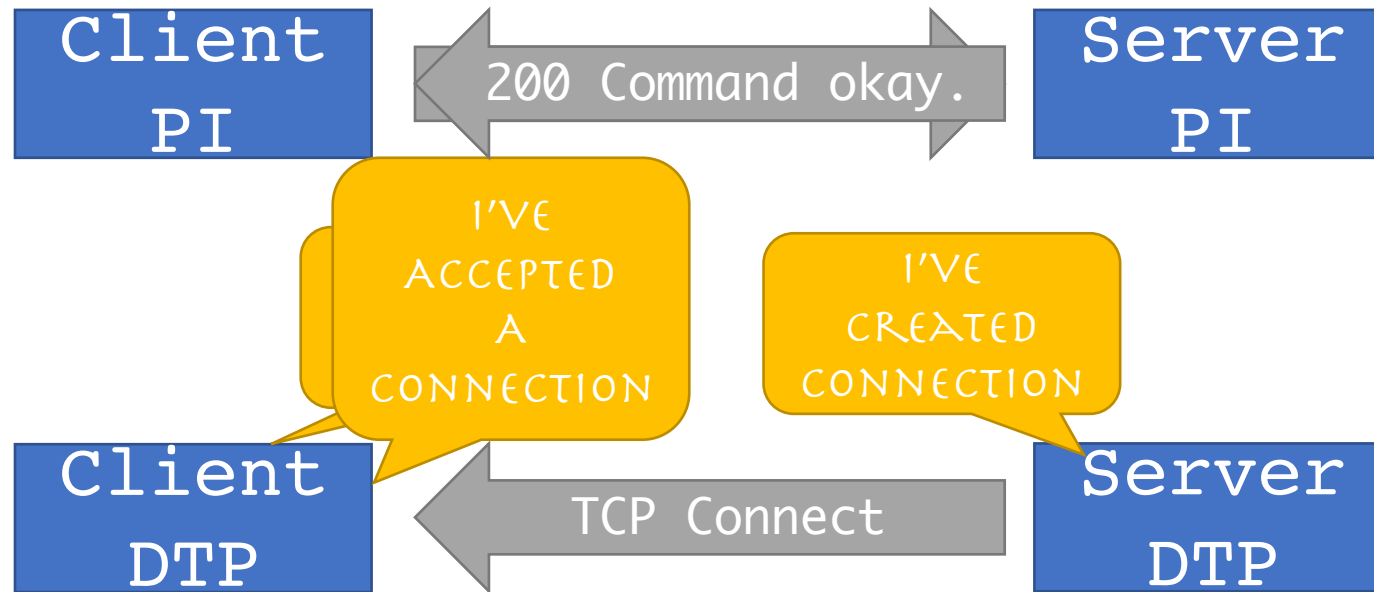


go learn()

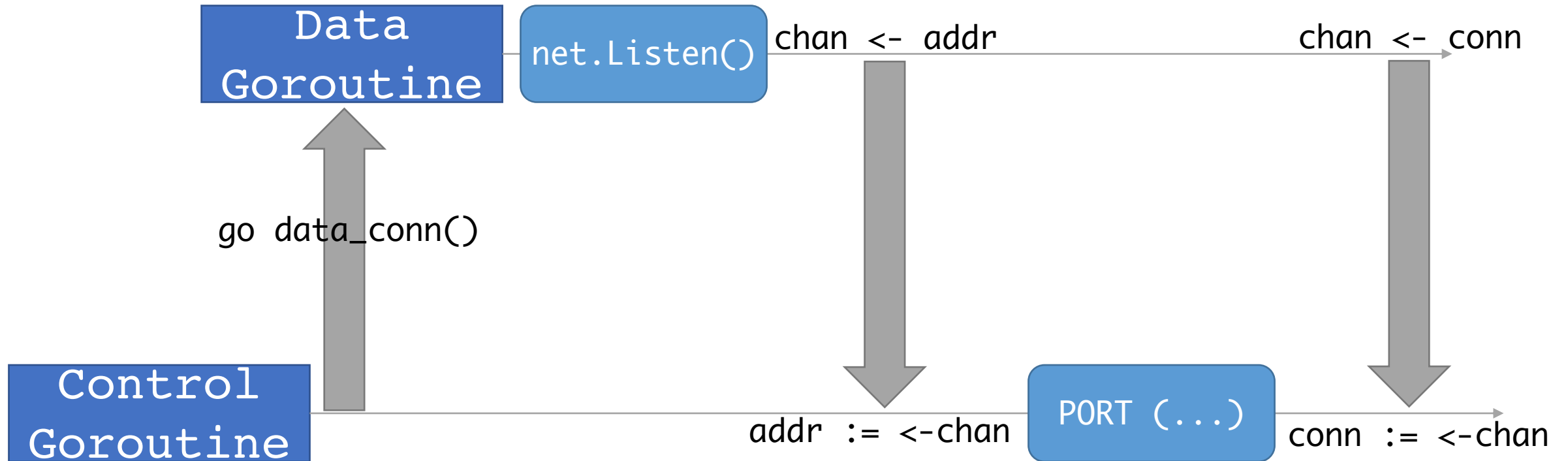


go sleep()

Example: Command PORT

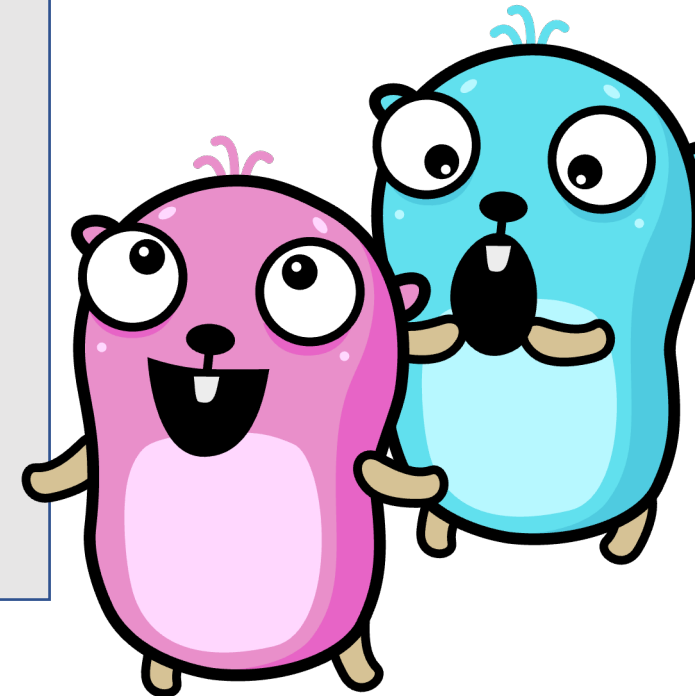
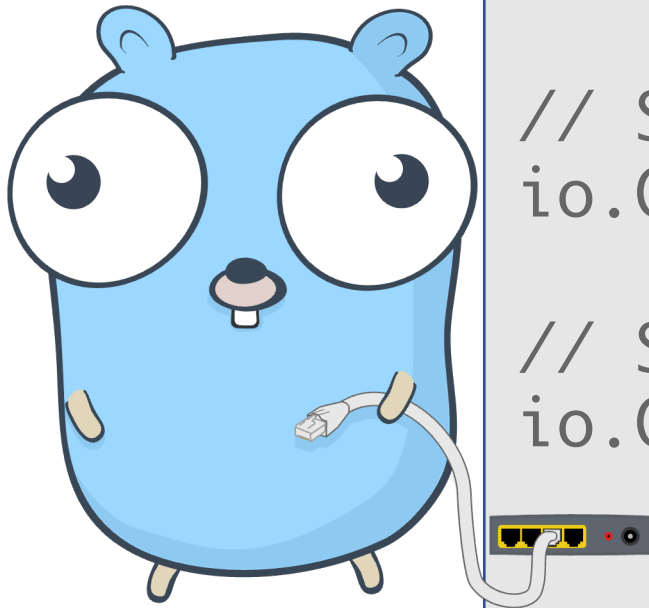


Example: Command PORT



IO and Network Packages

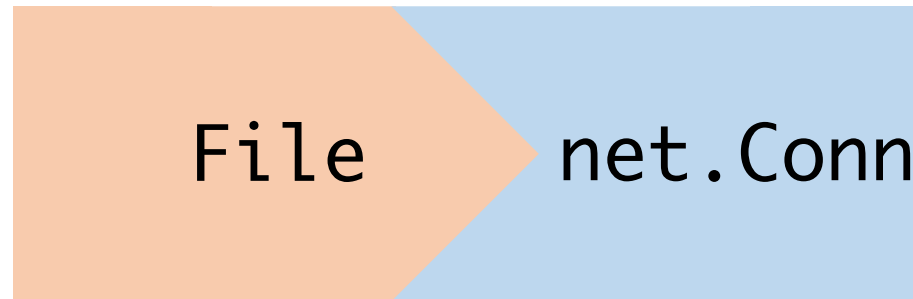
```
// Make a TCP connect  
conn, _ := net.Dial("tcp", addr)  
  
// Send a local file to connect  
io.Copy(f, conn)  
  
// Save conn data to local file  
io.Copy(conn, f)
```



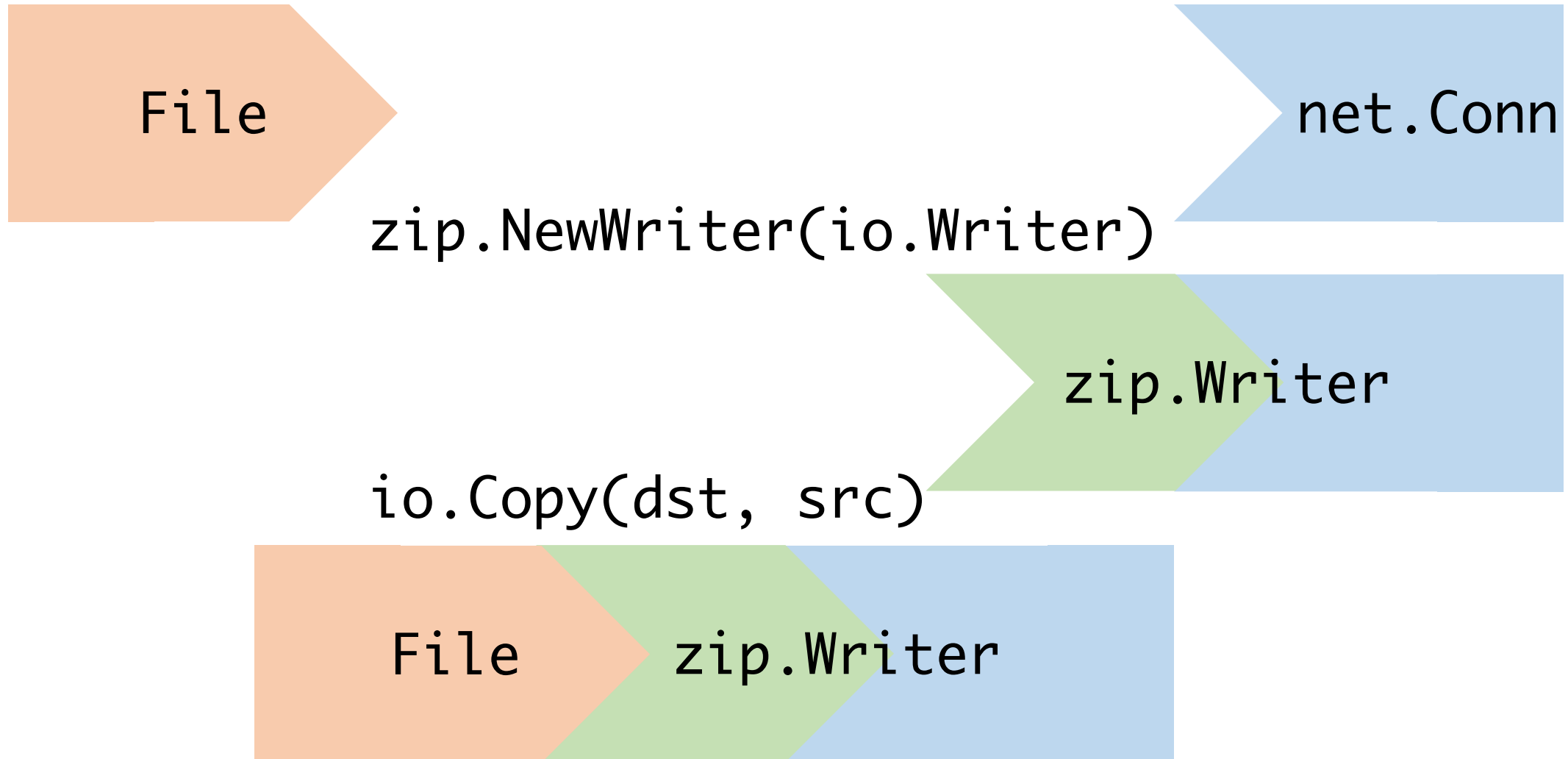
Stream Model



`io.Copy(dst, src)`



Example: Compressed Mode



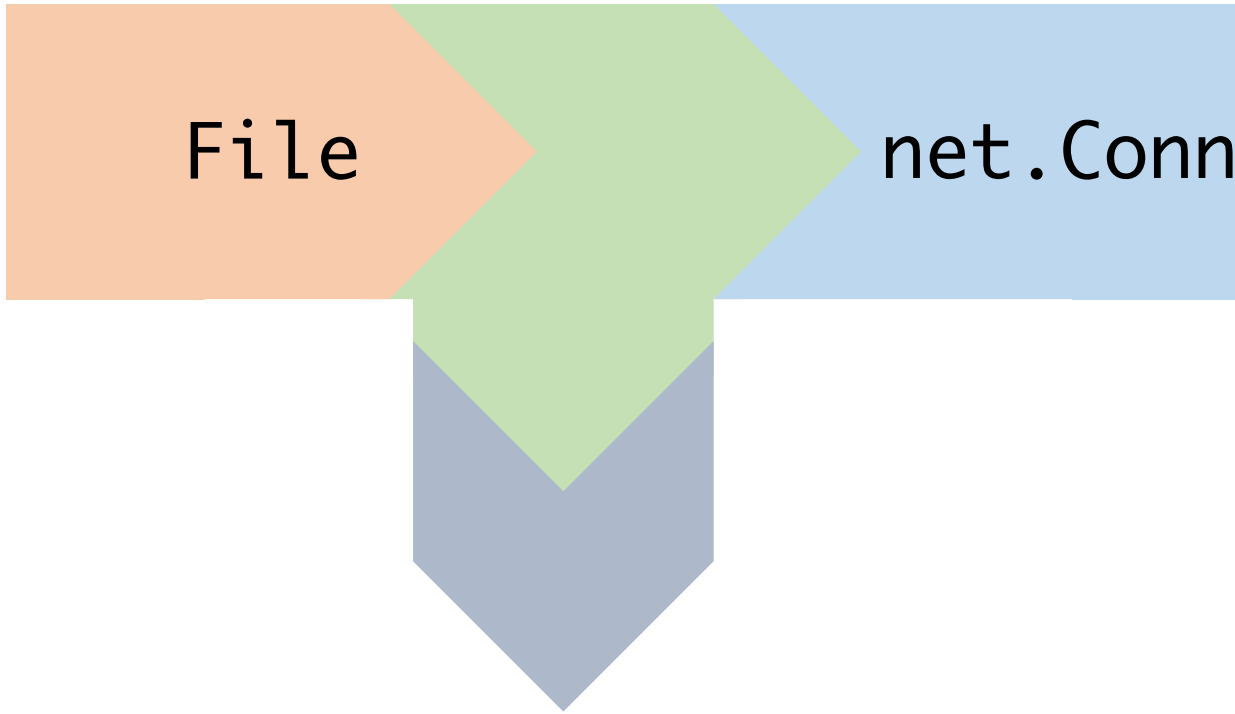
Example: Block Mode



```
r := io.TeeReader(dst, src)
```



Example: Block Mode



```
hash.Hasher()
```

```
io.CopyN(dst, src, n)
```

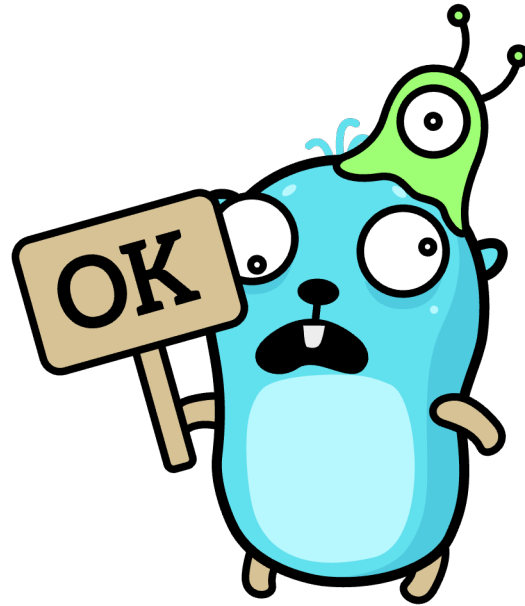
```
h := hasher.Hash()
```

```
net.Conn.Write(h)
```

```
hasher.Reset()
```

Pitfalls (and Solutions)

- Project Managers Differences
- Communication between Golang and Android



Dependencies Injection

```
// Golang
// FTP server should print
some log

type Logger interface{
    Log(string)
}

func NewFtpServer(Logger, ...)
```

```
// Java
public class MyLogger
implements server.Logger{
    @override
    public void log(String){...}
}

server.newFtpServer(
    new MyLogger(), ...)
```

Adapter Pattern

```
// Golang
// I want to read/write
// Android files

type File interface{
    Read(...)
    Write(...)
}
```

```
// Java
public class MyFile implements
server.File{
    public MyFile(File f){...}
    @override
    public read(...){
        return f.read(...);
    }
    @override
    public write(...){
        return f.write(...);
    }
}
```

Thanks! Let's GO!!

