

# Golang

By Mathias Nawacki & Fabian Witeczek

# Agenda

1. Introduction
2. Strengths of golang
3. Area of use
4. Cheatsheet
5. Assignments
6. Goroutines
7. More Assignments

# Introduction

- Garbage collected
- Strongly typed
- Compiled, not interpreted
- No classes (only structs)
- Pointers

# Strengths of golang

- Minimalistic
- Implicit interfaces
- Easy to read (`go fmt x.go`)
- Concurrency
- Official tools (documentation, code coverage)
- Open source
- Binaries without dependencies

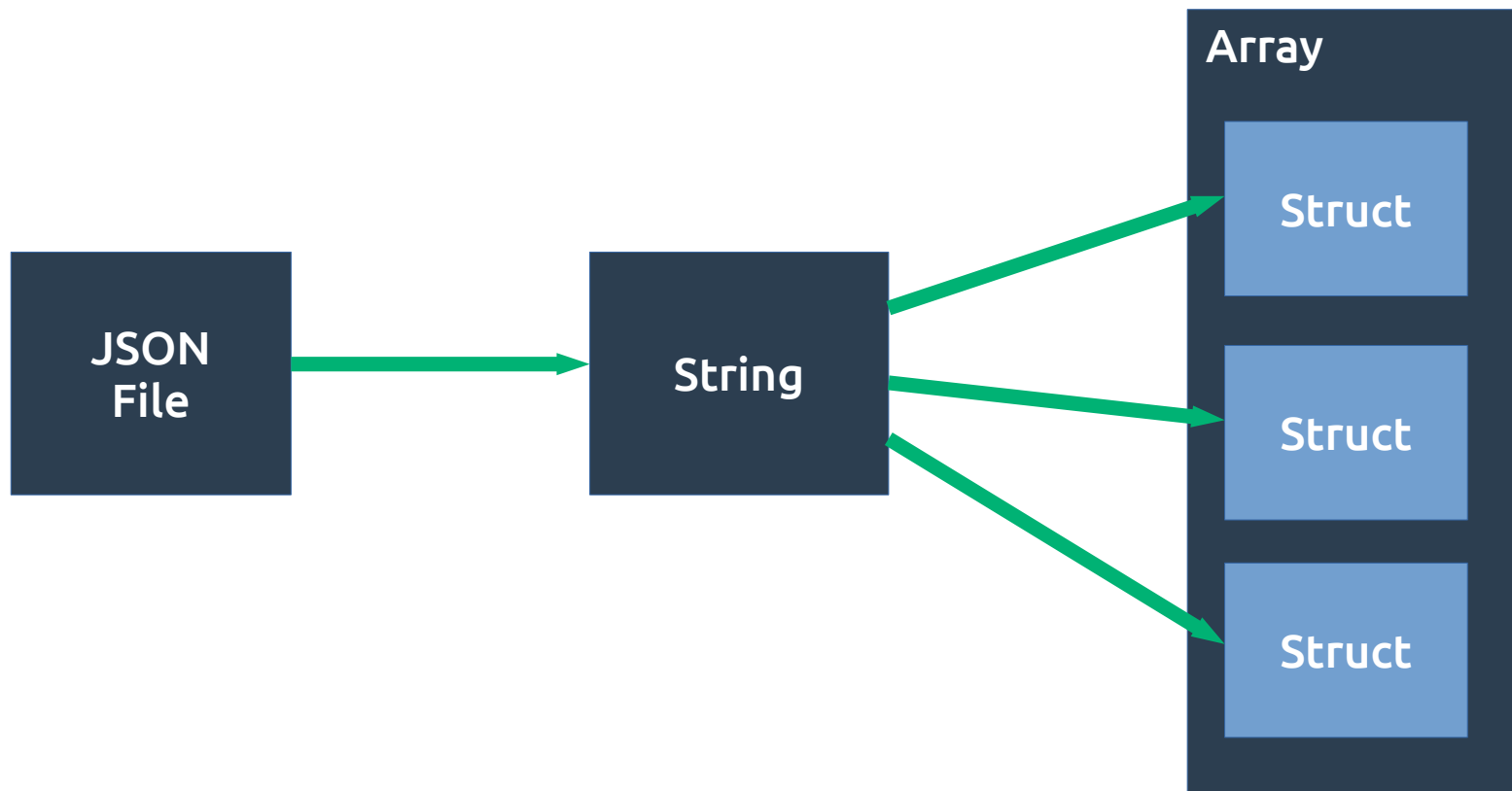
# Areas of use

- Networking & Web services
- Microservices
- Dataprocessing (I/O)



**Cheatsheet**

# Task 1: Reading a JSON file



## Task 2: Creating JSON string







**goroutines**

# Goroutines

- an independently executing function, launched by a go statement
- has its own call stack
- not a thread
-

# Goroutines

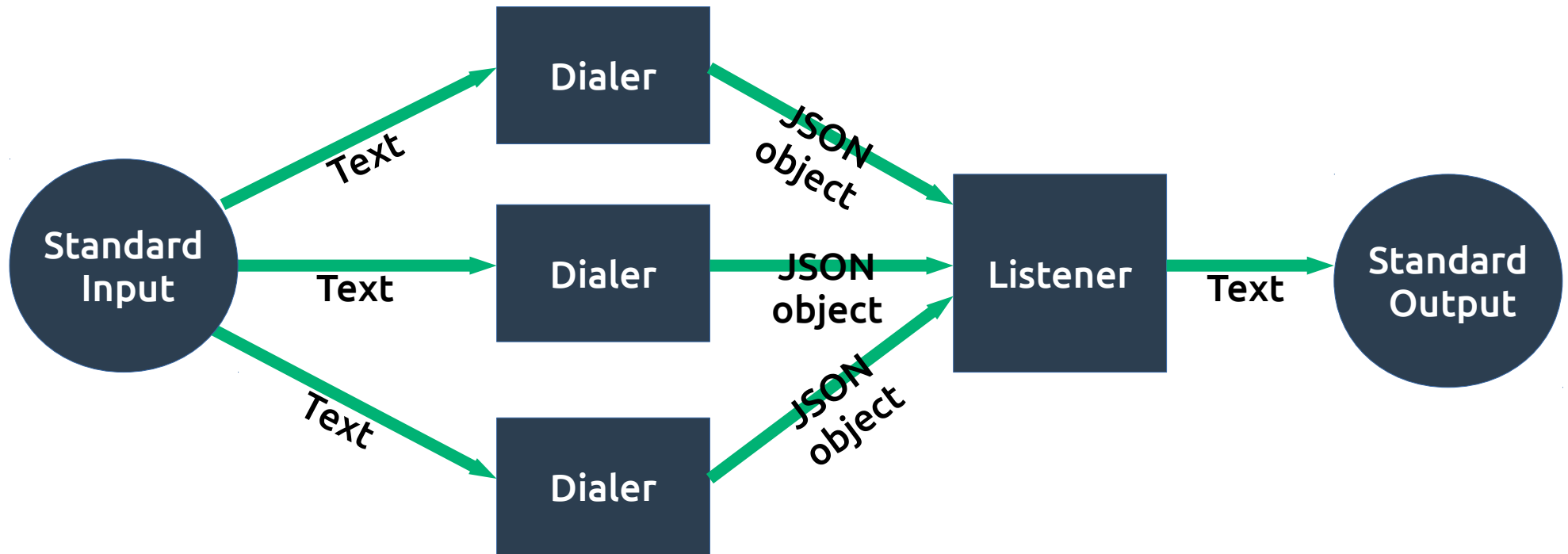
- **A Channel in Go provides a connection between two goroutines, allowing them to communicate**

```
//Declaring and initializing  
var c chan int  
c = make(chan int)  
// or  
c := make(chan int)
```

```
//Sending on a channel  
c <- 1  
//Receiving from a channel
```

```
value = <- c  
//The arrow indicates the direction of data flow
```

# Task 4: Listening & Dialing





The End?



The End!