How I became a Go developer

Luca Panziera - The Financial Times

Who I am

Started working for the FT 3 years ago

I was hired as **Java developer**

I spent **10 years** implementing applications in Java

Mainly in computer engineering research projects

FT

Why Java?

Java is a popular language:

- Many libraries
- Cross-platform
- Garbage collection
- Memory security

What people were doing at the FT when I joined

Implementing the new **Content platform**

Going towards **microservices**...

Each microservice was a **Java application running on a VM** in two FT data centers and AWS.

In parallel…
… **Implementing a hand-rolled “container orchestration framework”**

FT

One day our software architect said...

WE MUST USE GO!

Why?

Go is efficient!...
And we can implement everything in few milliseconds!

FT

Let’s take a look at Go…

What is Go?

A Programming Language:

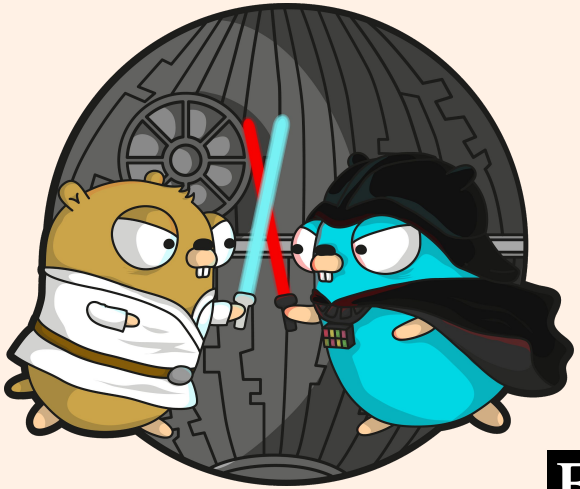
- developed by Google
- announced in November 2009
- Version 1.0 released in March 2012

So… it’s quite new!

FT

What Go developers had in mind?

- Statically typed and **scalable to large systems** (like Java or C++)
- **Simple and readable**, without excessive boilerplate
- **Not requiring IDEs**, but supporting them well
- Supporting **networking** and **multiprocessing**



FT

My first reaction to Go

```
import “math”

type Circle struct {
  x, y, r float64
}

func (c *Circle) Area() float64 {
    return math.Pi * c.r * c.r
}
```

Struct instead of Class?!

Methods are functions?!

Is this C?

FT

Why it looks like C?

Pointers

No overloading:

```
func addInt(a,b int) int { return a + b }
func addFloat64(a,b float64) float64 { return a + b }
```

Very basic data structures: Array, Slice, Map... nothing more

Go is a compiled language

FT

The reaction of my team...

We must use Go! It's going to make our life!

What the hell is this?

FT

Actually Go is really cool!

It is more concise… compared to Java…

Java

```
public class Circle {
  private double x, y, r;

  public double getX() { return x; }
  public void setX(double x) { this.x = x; }

  public double getY() { return y; }
  public void setY(double y) { this.y = y; }

  public double getR() { return r; }
  public void setR(double r) { this.r = r; }

  public double area() {
   return Math.PI * r * r;
  }
}
```

Go

```
import “math”

type Circle struct {
  X, Y, R float64
}

func (c *Circle) Area() float64 {
      return math.Pi * c.R * c.R
}
```

FT

Go is easy to learn

Go Tour

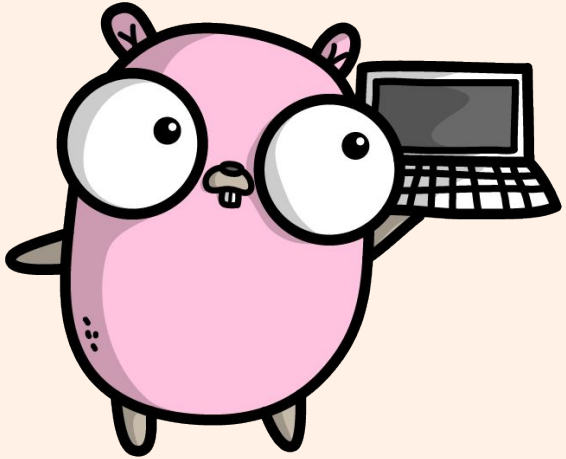
+

Effective Go

+

Bit of practice

After few days



Decent Go developer

FT

CPU and memory efficiency

- **Java microservice:**
  JSON manipulation,
  9K requests/day,
  340MB memory

- **Go microservice:**
  Database query + data manipulation,
  40K requests/day,
  12MB memory

Save cost of Cloud Services with smaller VM instances

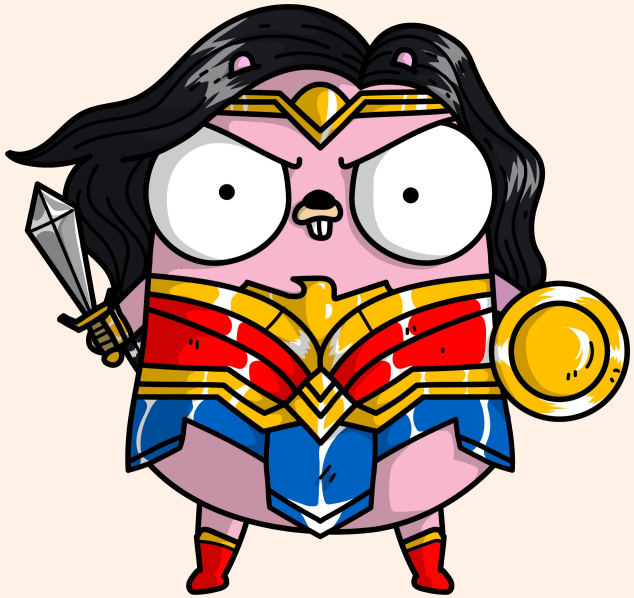
FT

What about memory allocation?!

Memory safety and Garbage collection

```
func brakeSlice() {
    a := []int{1, 2, 3}
    a[5] = 7
}
```

```
panic: runtime error: index out of
range
```

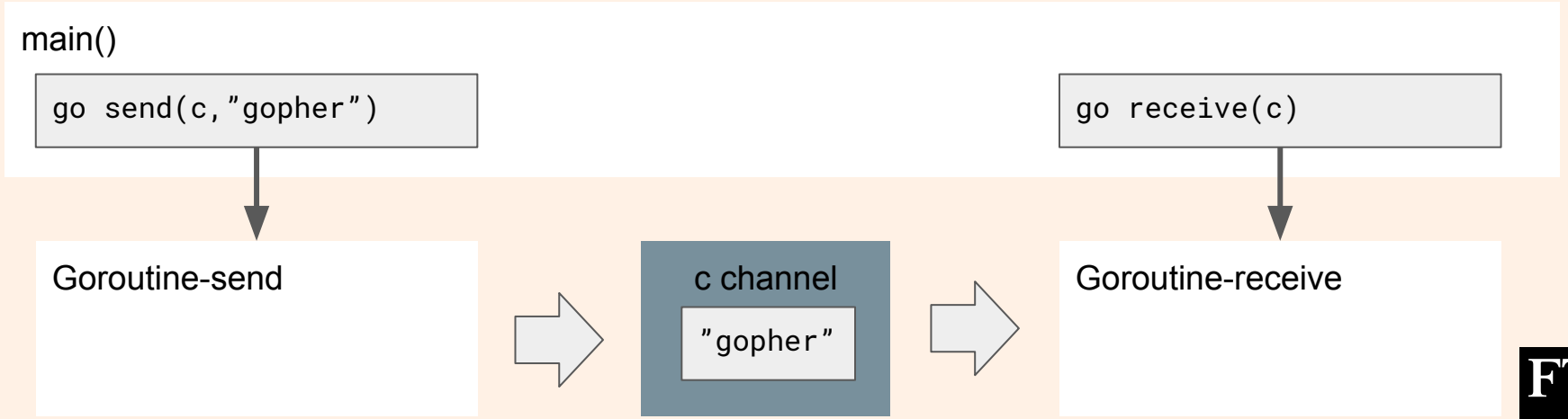


FT

A simple way to handle concurrency

```
func send(c chan string, msg string) {
     for {
          c <- msg
     }
}
```

```
func receive(c chan string) {
     for {
          msg := <-c
          fmt.Println(msg)
     }
}
```



No more Maven hell

Go is a compiled language so...
Fast in compiling and running test (few seconds)

A easy import of libraries with **“dep”**

```
dep init
dep ensure -add github.com/sirupsen/logrus
dep ensure -update

go test ./...

go build
```

FT

Go sounds amazing… but...

When the team started using Go...

People forgot about code practices!

FT

Everything in a single file

main.go

Usage of func: line 34

Usage of func: line 232

Impl. of func: line 476

Usage of func: line 675

A good excercise for head bagging

FT

Code in many files… but without any logic!

user.go

Usage of func

pluto.go

pippo.go

cat.go

record.go

Where is the implementation of func?

meaningless.go

Impl. of func

FT

Unit tests were not really mature...

Coverage wasn’t great

Difficulties in mocking

Missing Integration tests

Couple of times the worst happened in Production

FT

OMG… What's the situation now?

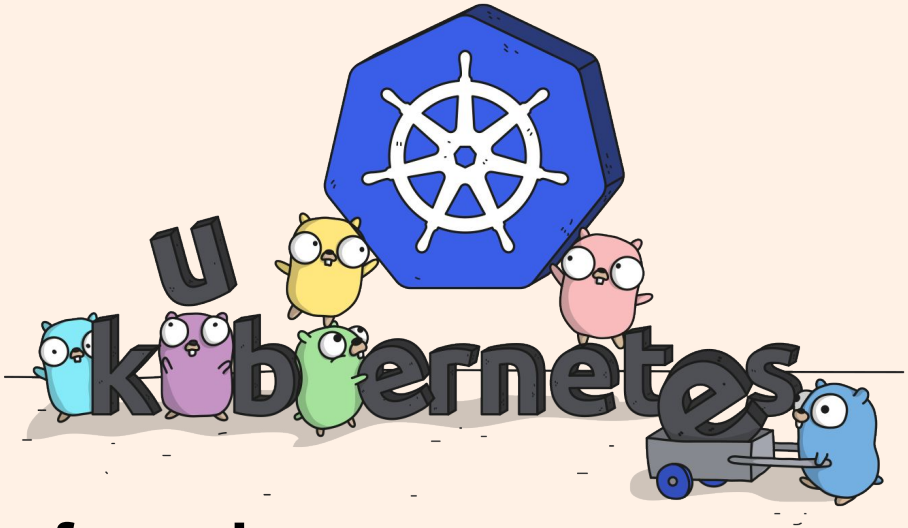
The FT Content platform

After 3 years... The quality of code base improved a lot

A Kubernetes-based platform made of **~150 microservices**

85% of them are implemented in Go

Reduced cost of 80% on AWS



So, **Go has been a success for us!**

FT

In the world of microservices…

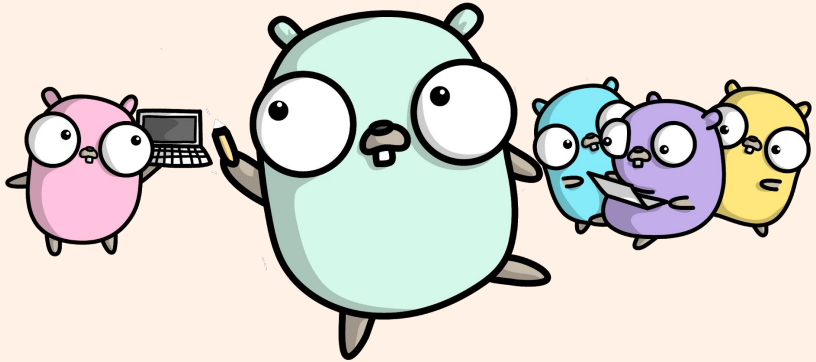
Fast implementation from scratch

Efficiency with network overhead

Go meets this requirements

FT

What I learnt by using Go?



Go combines:

- efficiency
- memory safety and garbage collection

A concise programming language

No language boundaries allow you to write excellent code… or total crap

Good sense of the developer is the key (as usual).

FT

Thank you!

We are hiring!
ft.com/dev/null

@izzyblues83

Credit Ashley McNamara and
https://github.com/ashleymcnamara/gophers for most images.
Credit Renee French for the original gopher concept and design.

FT