3 years a Go programmer

A talk by Elliott Stoneham at NorDev on 3rd February 2016

Any Gophers in the room?

Outline of this talk

- ‘Go & me’ some context
- The founding myth
- Lovers (and haters)
- At Google scale
- Concurrency examples
- Q&A

- 50 years ago: watched very first Dr Who episode
- 40 years ago: started programming (punch cards)
- 30 years ago: a systems programmer (C)
- 20 years ago: became an IT manager (OU MBA)
- 10 years ago: non-IT (FS,5* service, compliance)

我三年学习汉语。

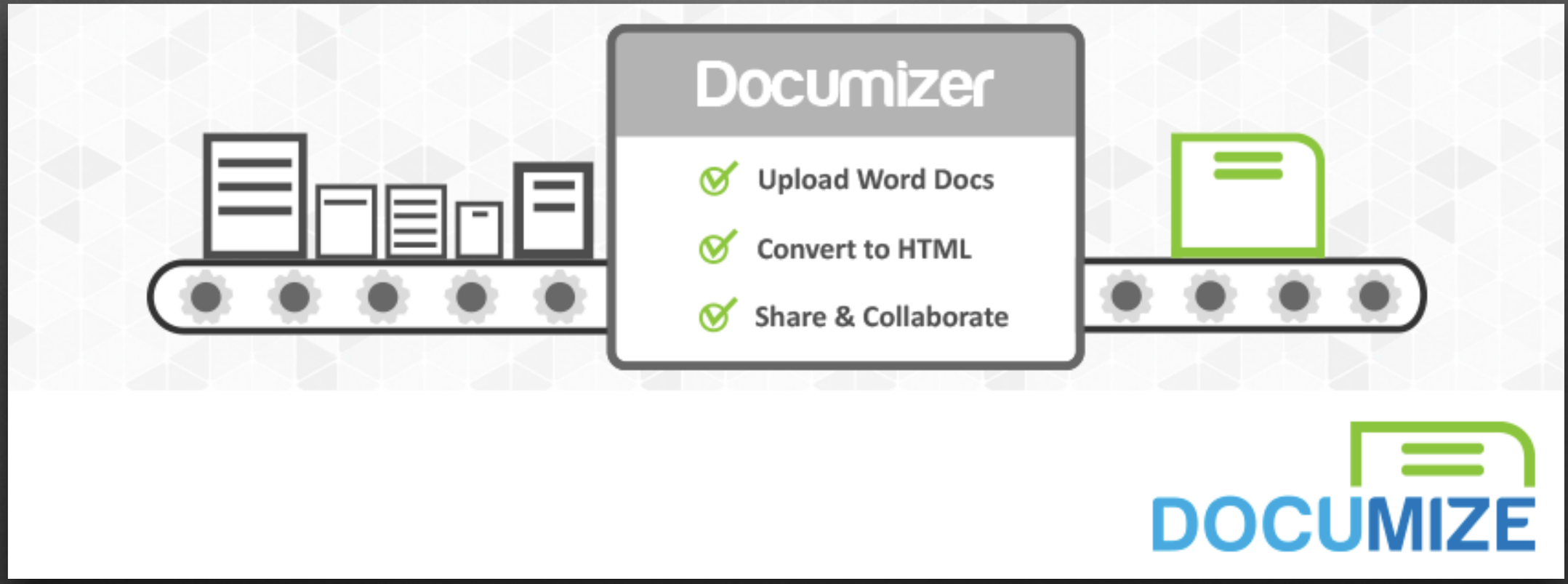
Master of Science

of this University

in Chinese Language, Business and International Relations

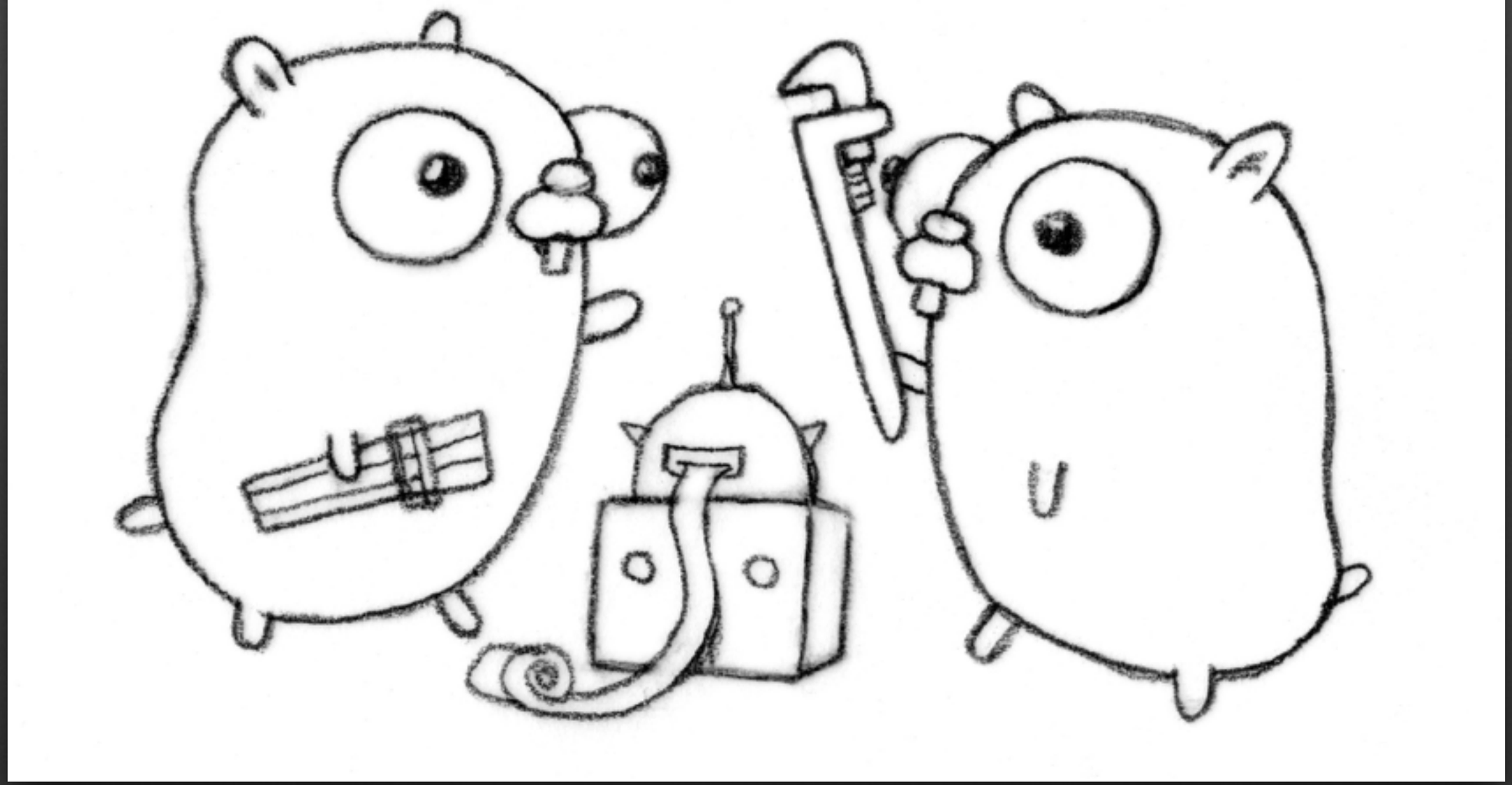
3 years ago I discovered Go…

[Gopher logo by Renée French]



www.documize.com

where I am a back-end developer



Go

Why create a new language?

Who were the founders?

- Ken Thompson (B, C, Unix, UTF-8)
- Rob Pike (Unix, UTF-8)
- Robert Griesemer (Hotspot, JVM)

...and a few others engineers at Google

[http://www.slideshare.net/suelengc/go-lang-52138194]
[other slides also used - thank you Suelen Goularte Carvalho!]

Waiting for a C++ compilation

[cartoon at http://xkcd.com/303/]

the birth of Go in 2007

“When the three of us got started, it was pure research. The three of us got together and decided that we hated C++.

...we started off with the idea that all three of us had to be talked into every feature in the language, so there was no extraneous garbage put into the language for any reason.”

—Ken Thompson

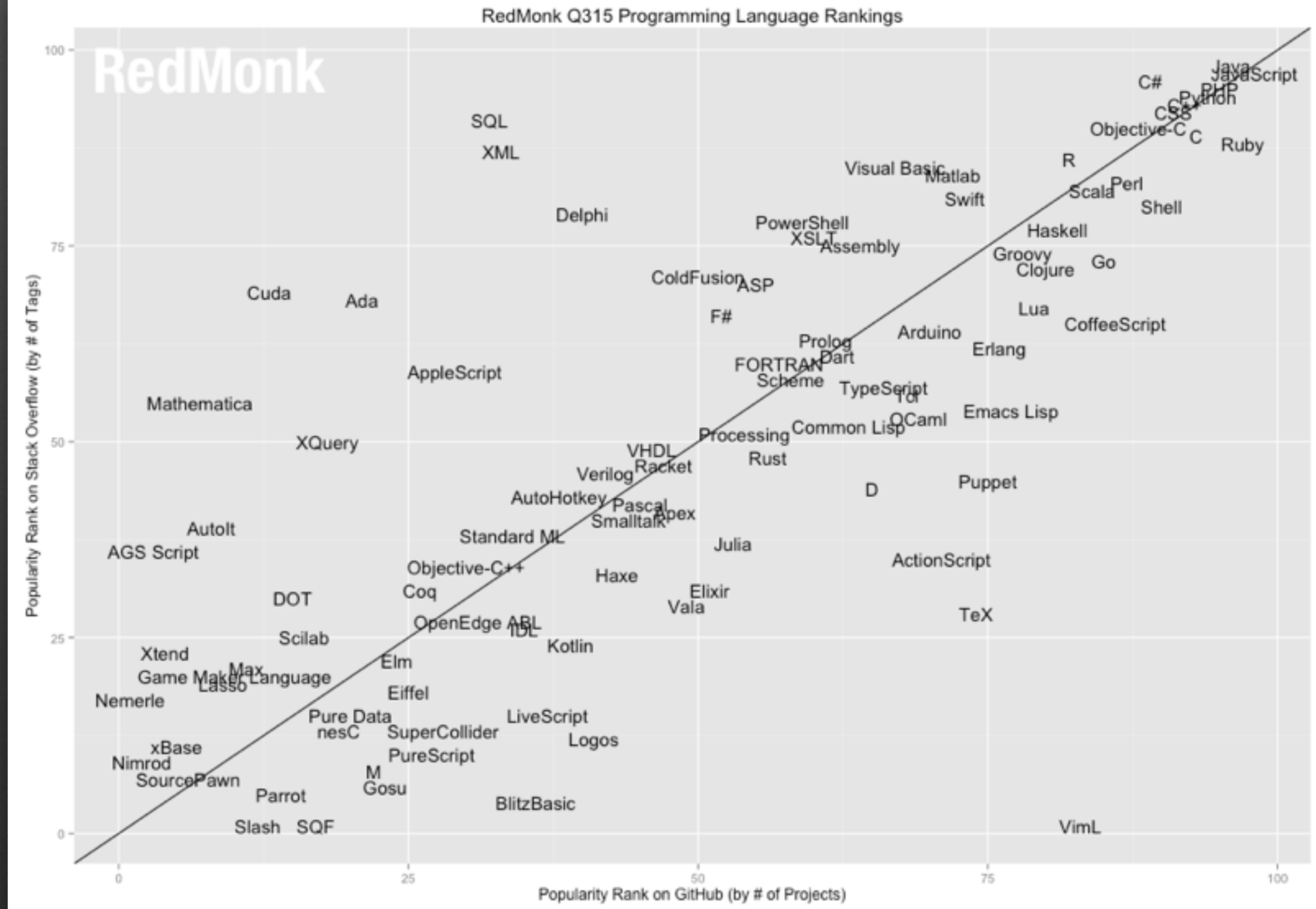
the objectives of Go

- *“Readable, safe, and efficient code.*
- *A build system that scales.*
- *Good concurrency support.*
- *Tools that can operate at Google-scale."*

— Robert Griesemer

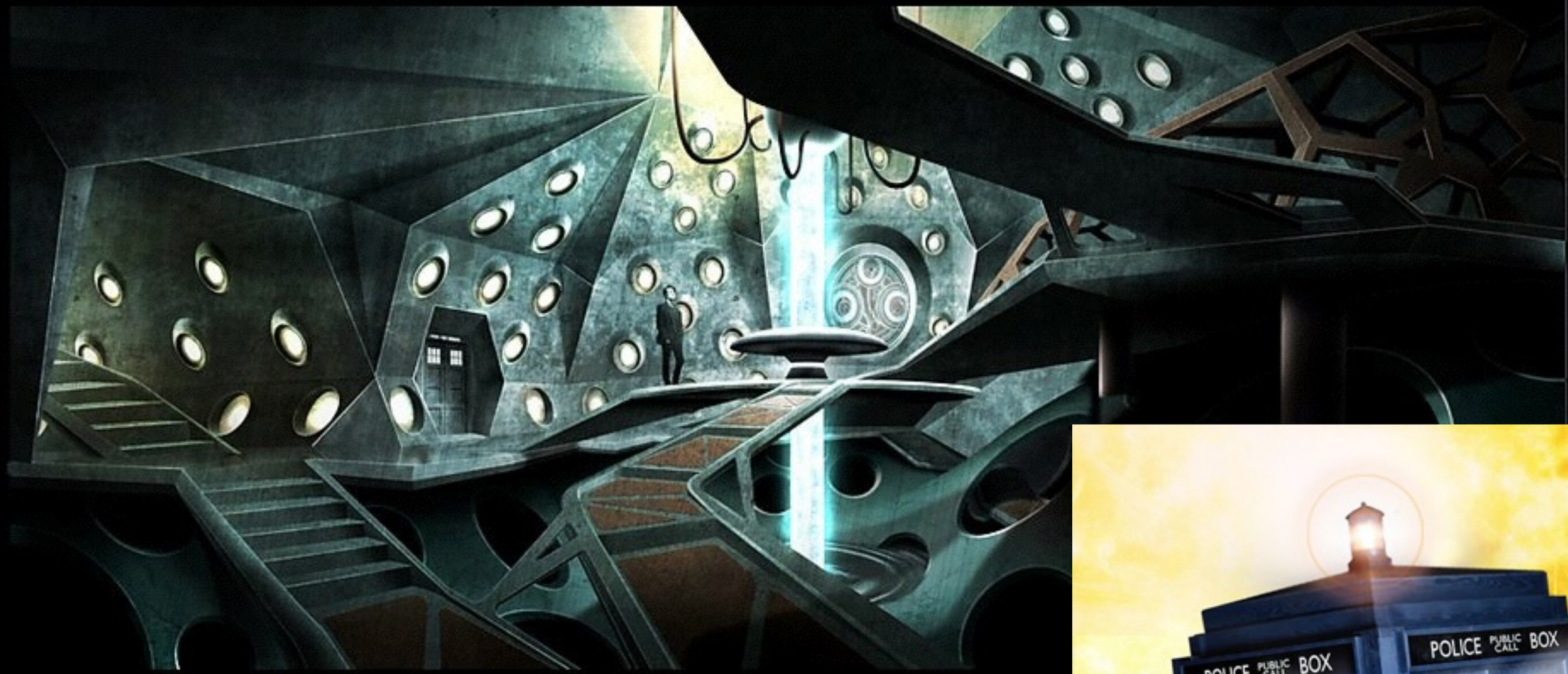
“Now, Go makes much more sense for the class of problems that C++ was originally intended to solve.”

–Bruce Eckel, author and
founding member of the ANSI/ISO C++ standard committee



Go is #15

[http://redmonk.com/sogrady/2015/07/01/language-rankings-6-15/]



Go is like the TARDIS

A small idiosyncratic exterior,
conceals large-scale quality engineering.

[TARDIS name & Dr Who images (c) BBC]

What will you see in Go?

- ❏ Compiled
- ❏ Garbage-collected
- ❏ Has your own runtime
- ❏ Simple syntax
- ❏ Great standard library
- ❏ Cross-platform
- ❏ Object Oriented (without inheritance)
- ❏ Statically and stronger typed
- ❏ Concurrent (*goroutines*)
- ❏ Closures
- ❏ Explicity dependencies
- ❏ Multiple return values
- ❏ Pointers
- ❏ and so on...

What will you not see in Go?

- ❑ Exception handling
- ❑ Inheritance
- ❑ Generics
- ❑ Assert
- ❑ Method overload

Have not been implemented in favor of efficiency.

Why Everyone Hates Go

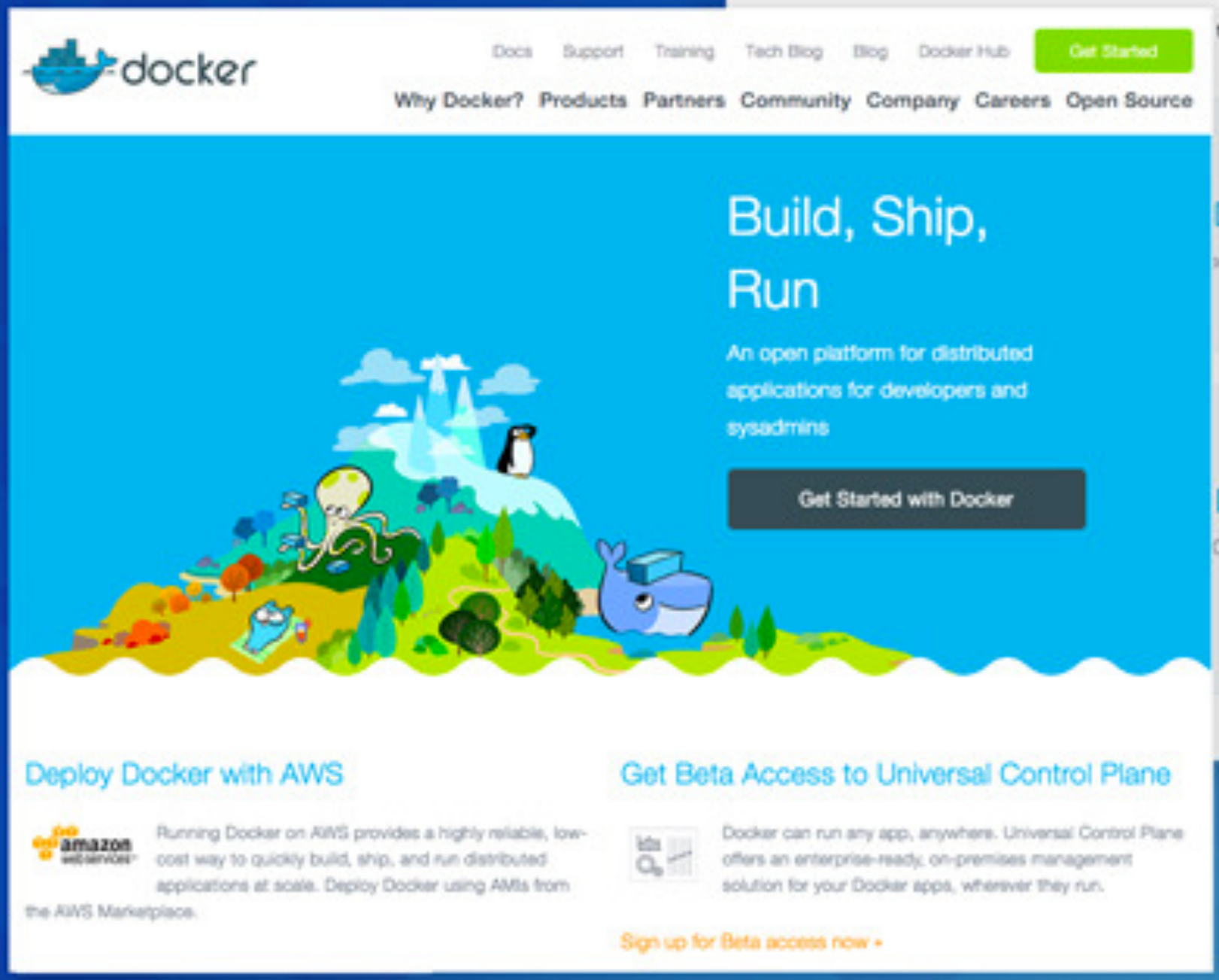
- Go is a language stuck in the 70's.
- Go ignores 40 years of programming language research.
- Go is a language for blue collar (mediocre) developers.

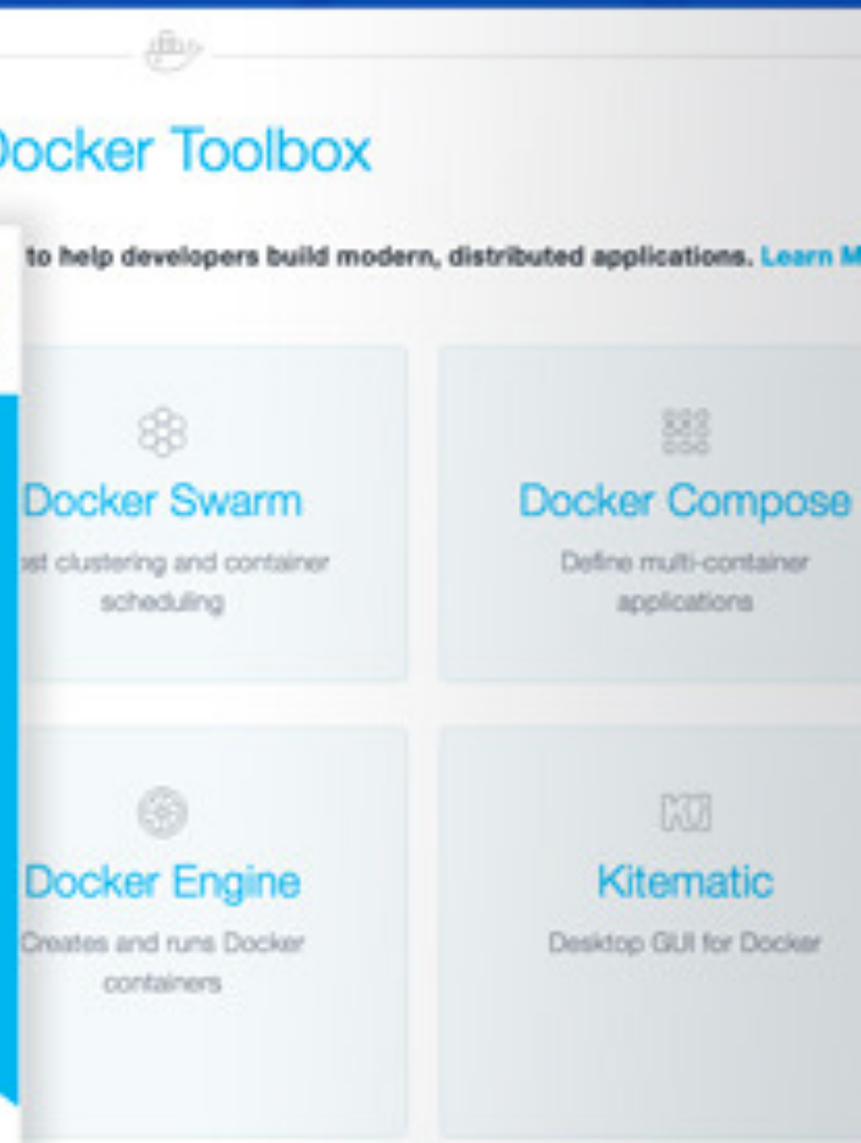
[http://npf.io/2014/10/why-everyone-hates-go/]

Write in Go (ScaleAbility)
[https://www.youtube.com/watch?v=LJvEIjRBSDA]

2016 Technology of the Year Award winners

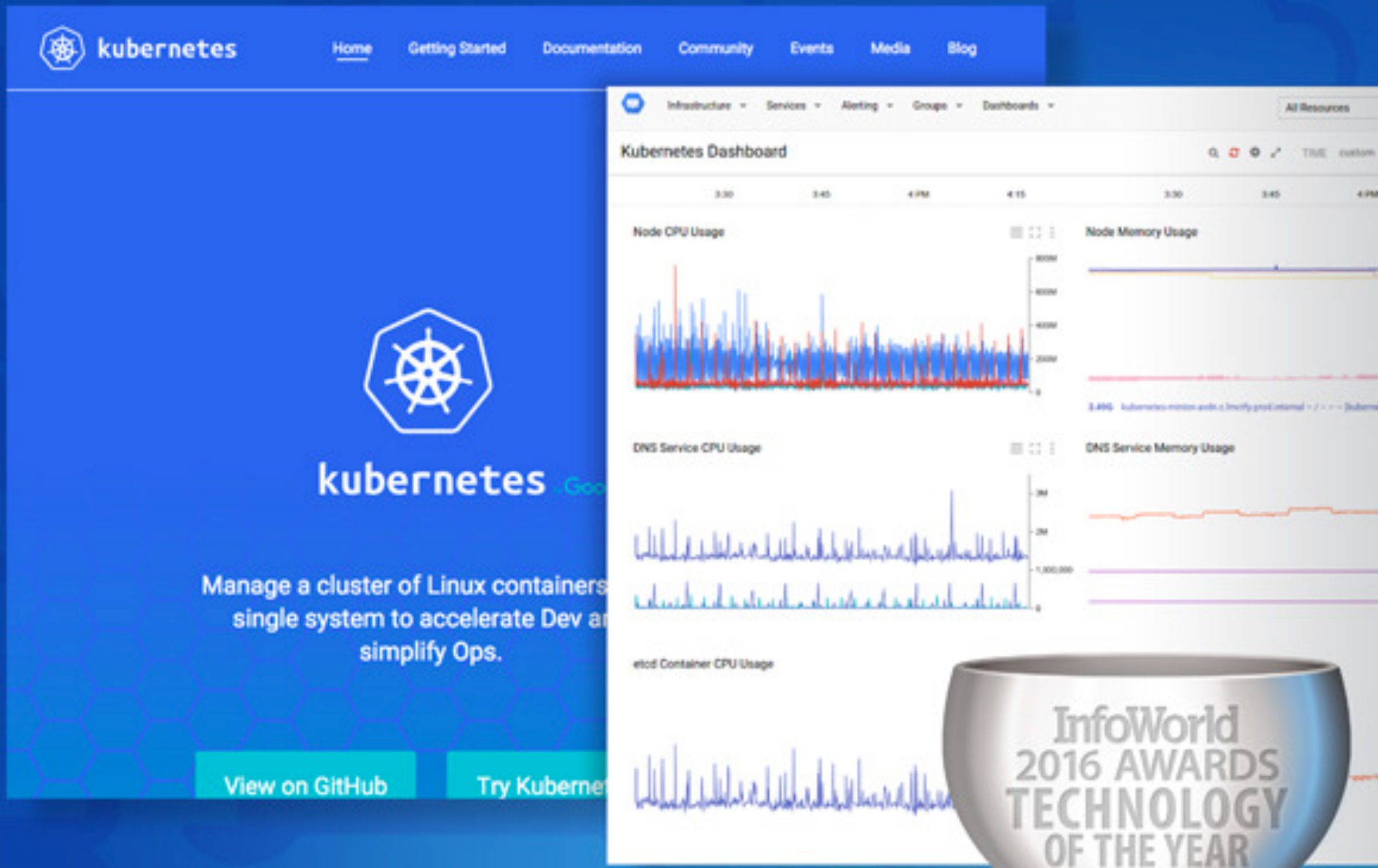
[http://www.infoworld.com/article/3023050/open-source-tools/infoworlds-2016-technology-of-the-year-award-winners.html]


docker
Docs Support Training Tech Blog Blog Docker Hub
Get Started
Why Docker? Products Partners Community Company Careers Open Source
Build, Ship, Run
An open platform for distributed applications for developers and sysadmins
Get Started with Docker
Deploy Docker with AWS
Running Docker on AWS provides a highly reliable, low-cost way to quickly build, ship, and run distributed applications at scale. Deploy Docker using AMIs from the AWS Marketplace.
Get Beta Access to Universal Control Plane
Docker can run any app, anywhere. Universal Control Plane offers an enterprise-ready, on-premises management solution for your Docker apps, wherever they run.
Sign up for Beta access now »

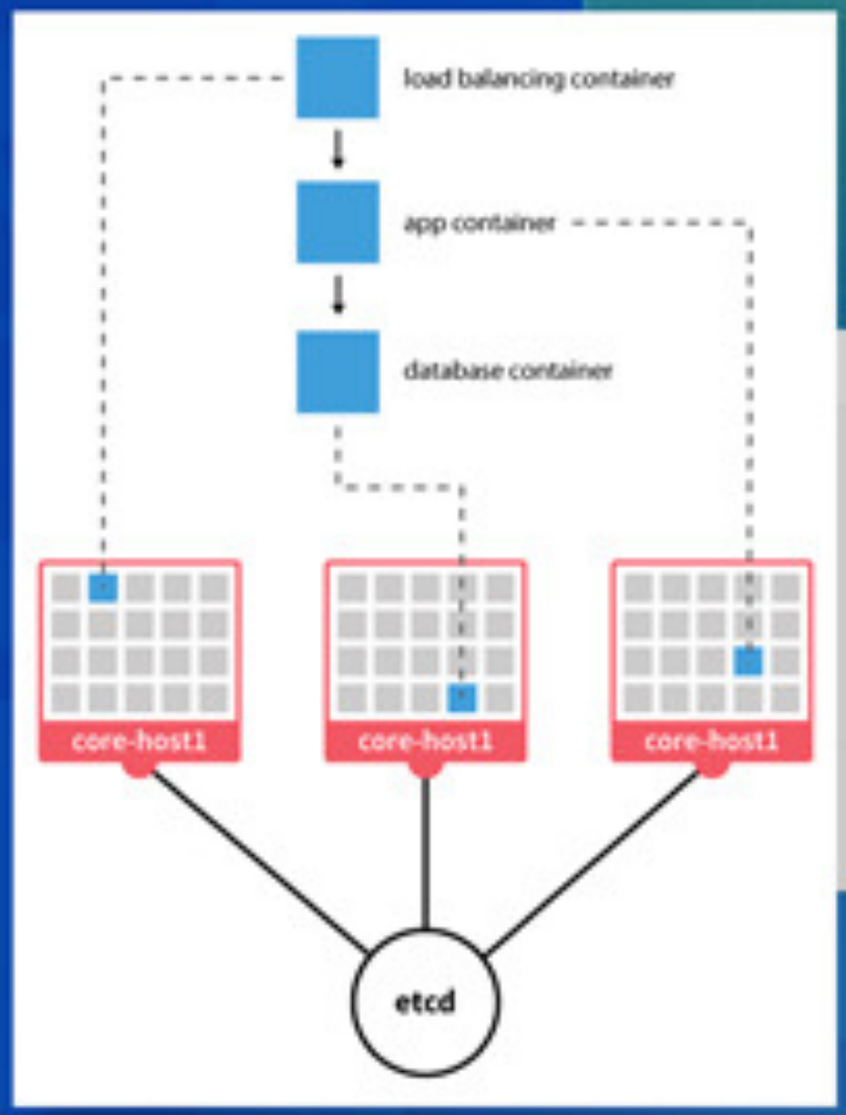

Docker Toolbox
to help developers build modern, distributed applications. Learn M
Docker Swarm
st clustering and container scheduling
Docker Compose
Define multi-container applications
Docker Engine
Creates and runs Docker containers
Kitematic
Desktop GUI for Docker


InfoWorld
2016 AWARDS
TECHNOLOGY
OF THE YEAR

Docker


kubernetes
Home
Getting Started
Documentation
Community
Events
Media
Blog
kubernetes
Manage a cluster of Linux containers
single system to accelerate Dev a
simplify Ops.
View on GitHub
Try Kuberne
Infrastructure
Services
Alerting
Groups
Dashboards
All Resources
Kubernetes Dashboard
Node CPU Usage
Node Memory Usage
DNS Service CPU Usage
DNS Service Memory Usage
etcd Container CPU Usage
InfoWorld
2016 AWARDS
TECHNOLOGY
OF THE YEAR

Kubernetes



CoreOS

Home Products Using CoreOS Documentation Blog

Open Source Projects for Linux Containers

Try out CoreOS

Latest version is CoreOS 899.1.0
Release Notes » Update Philosophy »

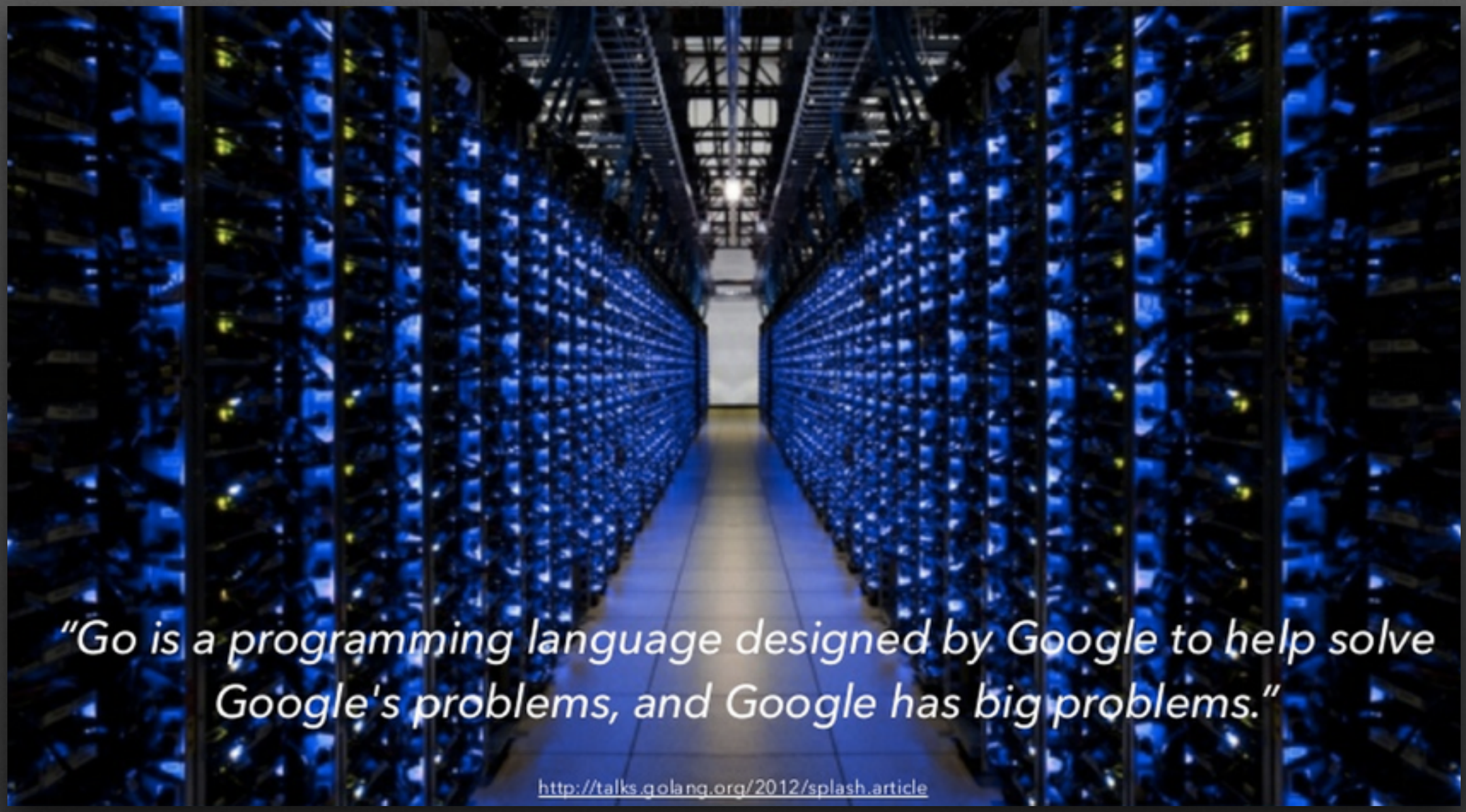
TECTONIC The Universal Kubernetes Solution, Tectonic, is now generally available

Popular open source projects for distributed apps:

CoreOS — OPERATING SYSTEM

etcd — CONSENSUS & DISCOVERY

rkt — CONTAINER RUNTIME



Software engineering in the LARGE

Complex distributed systems, running at scale; and written by big teams.

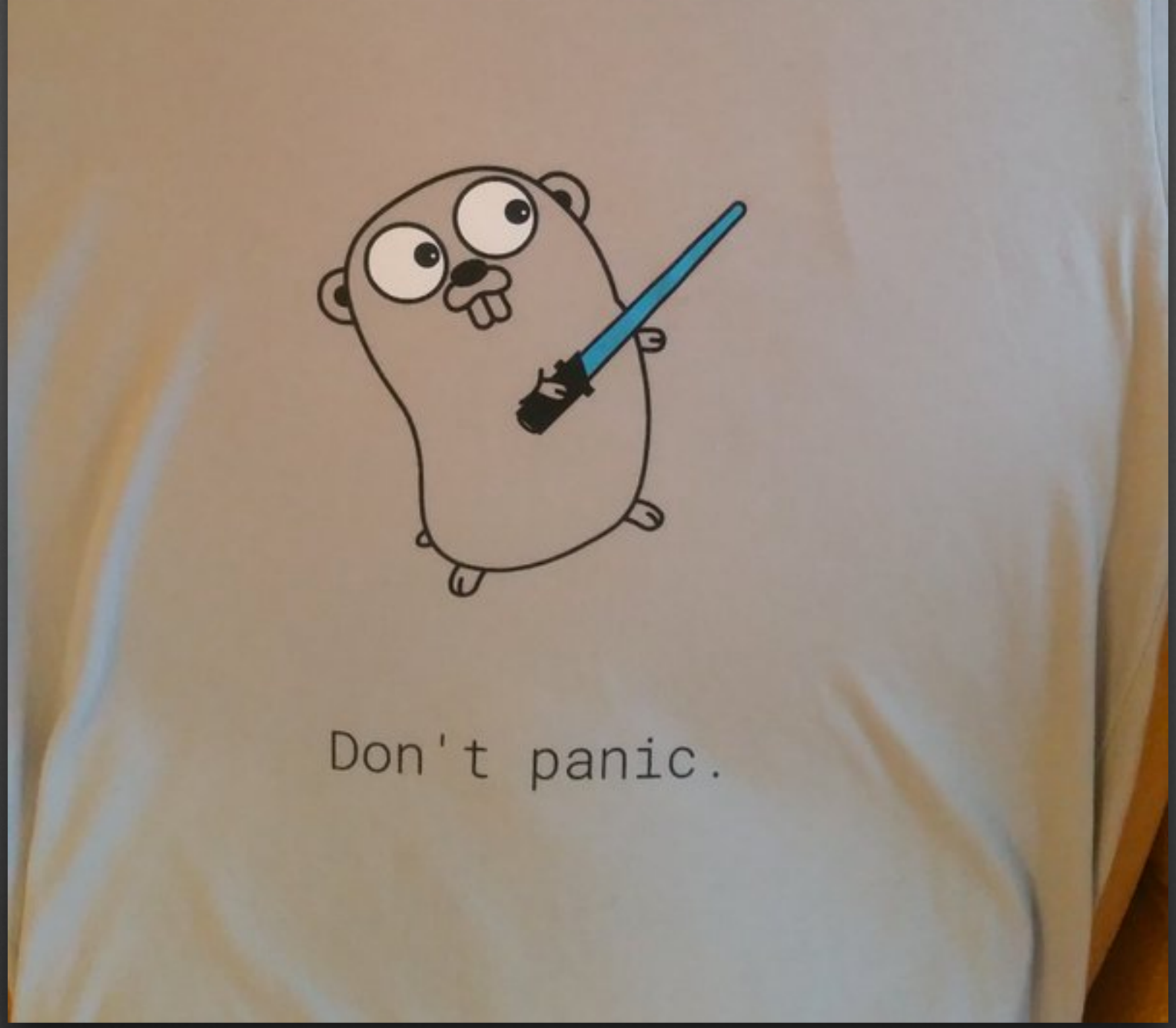
TL;DR

COMPLEX

SIMPLE

[A Recovering Java Developer Learns to Go]

[http://www.slideshare.net/mstine/java-devlearnstogooscon]



Simplicity is Complicated

go-proverbs.github.io

Complexity is time
and time is money …
both for humans and computers

Using Go you do not defer complexity cost, you take the hit up-front
… but it turns out that is usually more efficient overall

Development speed

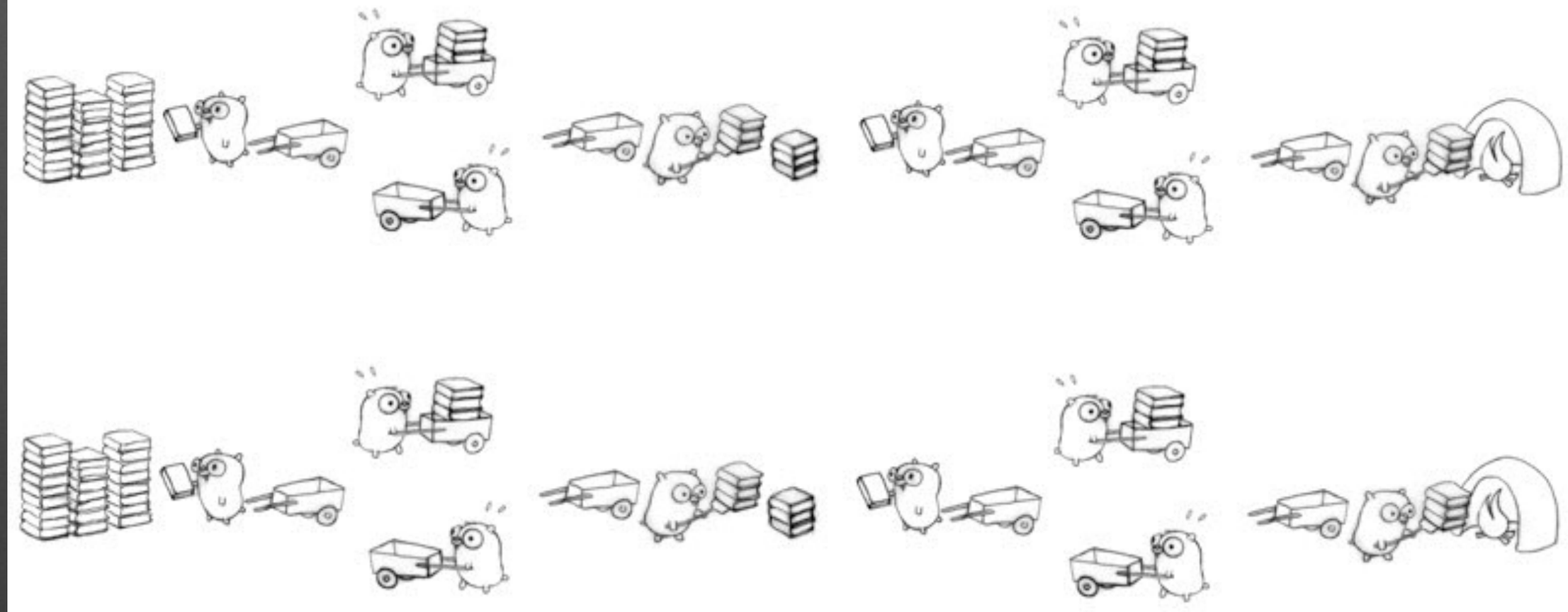
- Fast compilation, saving developer time (attracting NodeJS developers, rubyists & pythonistas)
- Speed of reading code (Go is small & understandable)
- Speed of refactoring (Go interfaces are very pragmatic)
- Speed of training new recruits (“blue-collar” devs)

Easy deployment

- Static binaries => very simple individual deployment
- Fast spin-up of additional servers, compared with VMs
- Cross-compilation => multi-platform deployment
- Robust networking library & built-in web server (http2)

Low cloud costs

- Go serves webpages roughly 100x faster than PHP
  [http://www.cyb.co.uk/blog/development/2014/07/why-you-should-consider-golang-for-your-next-web-application]
- Easy parallelism enables elapsed-time speed-ups on a single multi-core machine
- Low memory footprint compared with VMs
- Efficient distributed processing enables “big data” applications across networks of machines



Concurrency is not parallelism

image from a talk by Rob Pike
http://blog.golang.org/concurrency-is-not-parallelism

Concurrency (goroutines)

- To execute a goroutine, just **go**!
- To send or receive information between the goroutines, use **channels**
- Use the **GOMAXPROCS** environment variable to define the amount of threads

Goroutines

- A goroutine is a lightweight thread managed by Go runtime

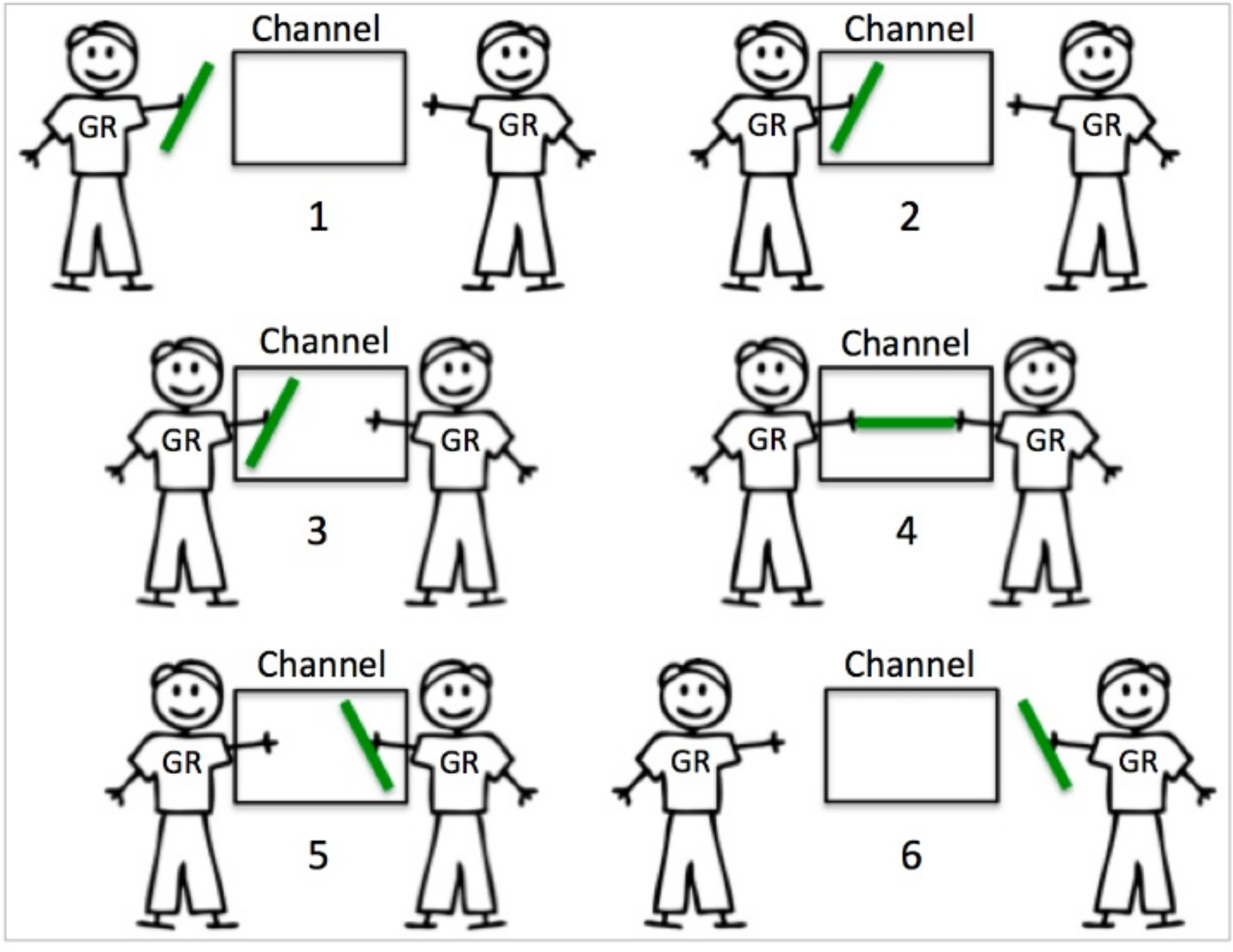
goroutines.go

```go
package main

import (
    "fmt"
    "time"
)

func say(s string) {
    for i := 0; i < 5; i++ {
        time.Sleep(100 * time.Millisecond)
        fmt.Println(s)
    }
}

func main() {
    go say("world")
    say("hello")
}
```

```
$ go run goroutines.go
hello
world
hello
world
hello
world
hello
world
hello
hello
```

Unbuffered Channels

```
c := make (chan int)
```



http://www.goinggo.net/2014/02/the-nature-of-channels-in-go.html

Channels

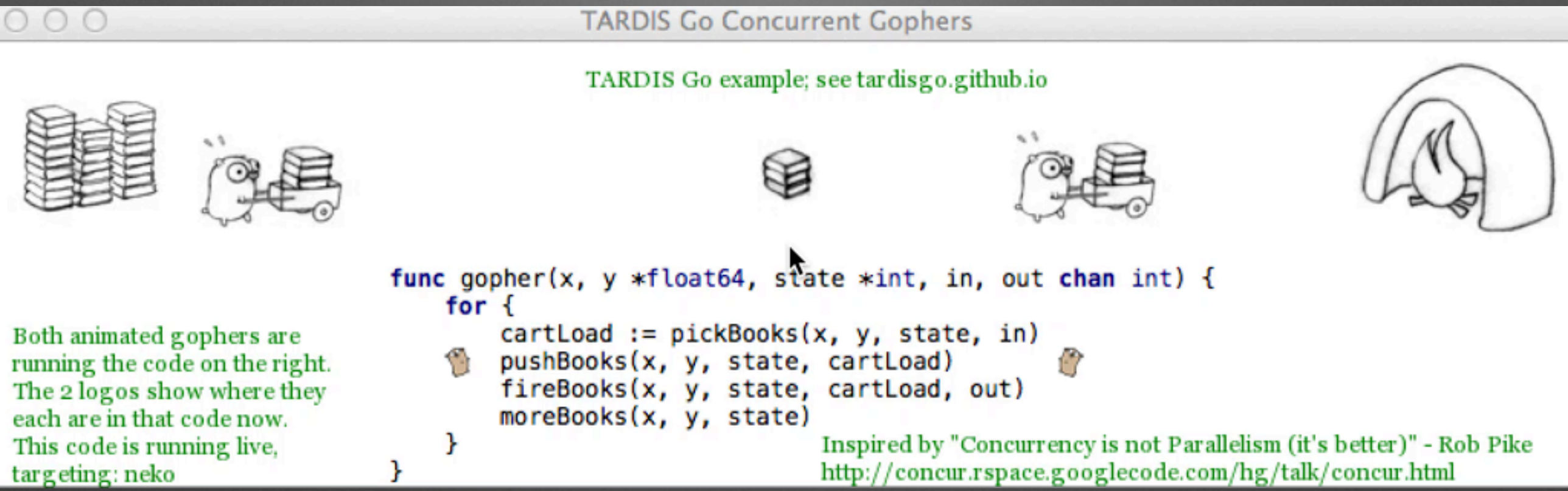
- Channels are typed's conduit through which you can send and receive values with the channel operator <-

channels.go

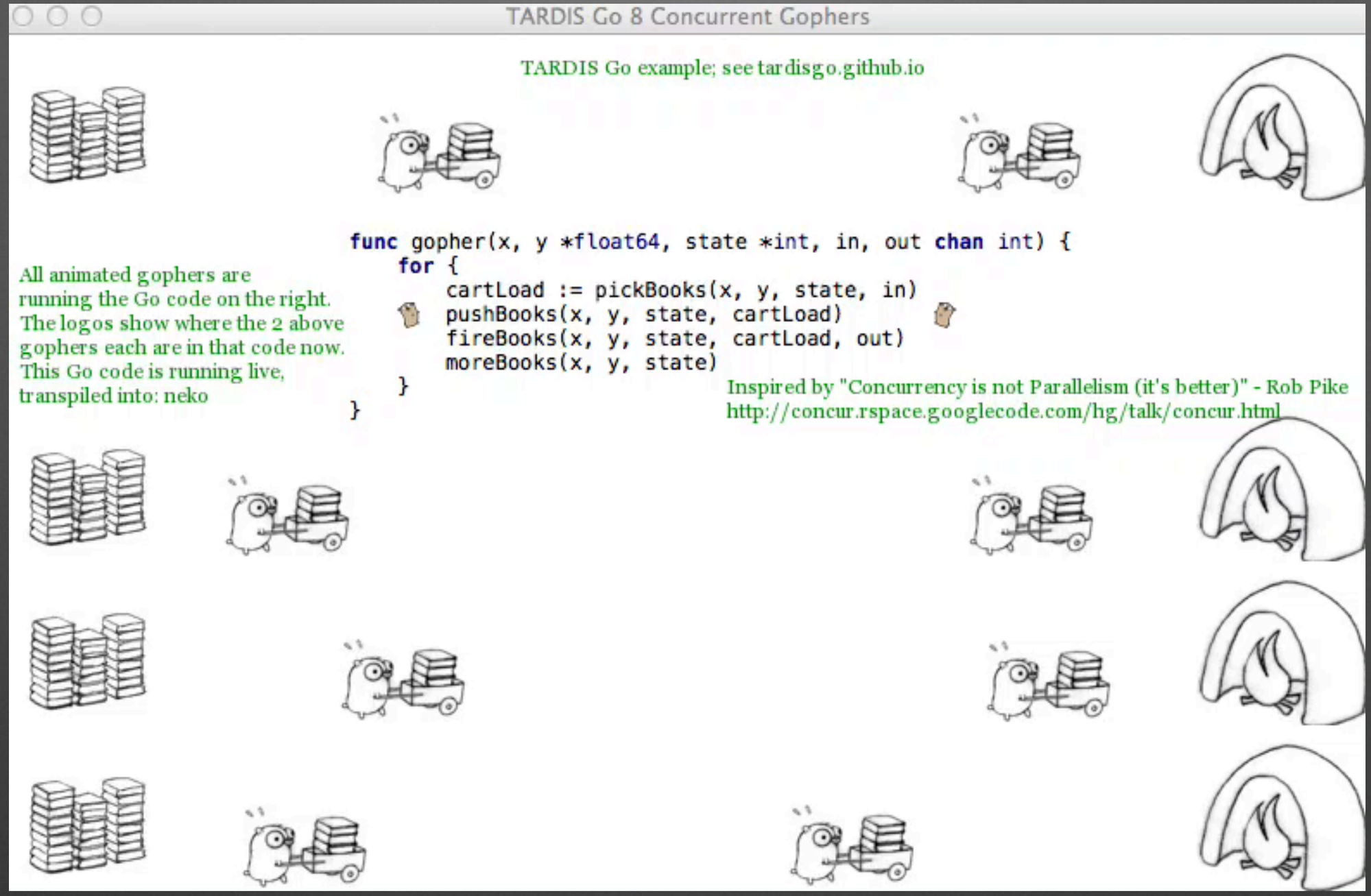
```go
package main

import "fmt"

func sum(a []int, c chan int) {
    sum := 0
    for _, v := range a {
        sum += v
    }
    c <- sum // send sum to c
}

func main() {
    a := []int{7, 2, 8, -9, 4, 0}

    c := make(chan int)
    go sum(a[:len(a)/2], c)
    go sum(a[len(a)/2:], c)
    x, y := <-c, <-c // receive from c

    fmt.Println(x, y, x+y)
}
```

```
$ go run channels.go
17 -5 12
```

tardisgo.github.io

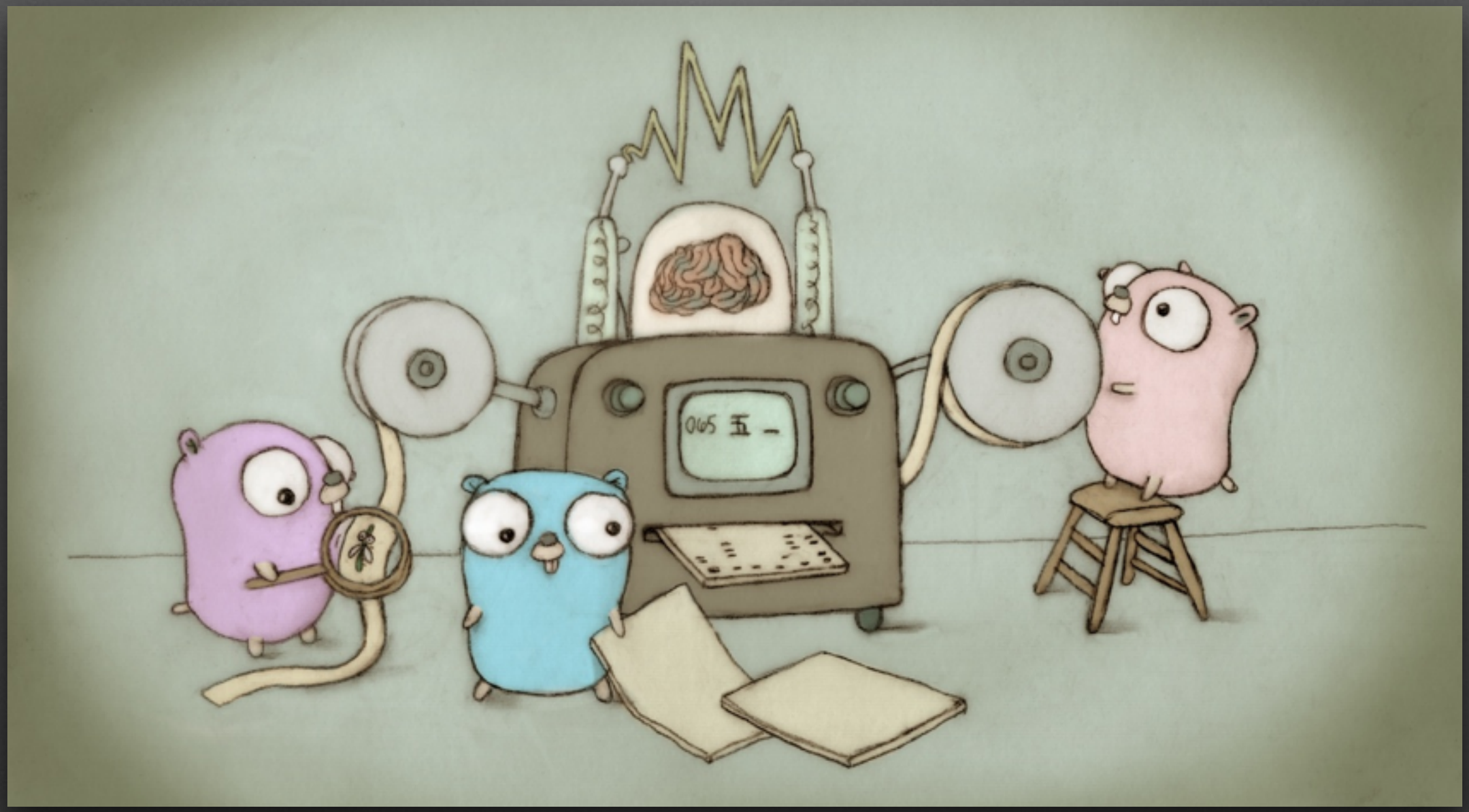


- Each of the two sprites is a Go “goroutine"
- The pile of books in the middle is a Go “channel”
- This is pipeline concurrency (and parallelism if >1 CPU)



burn those C++ manuals!

(TARDIS Go —> Haxe + Haxe / OpenFL; running on the neko target)



$ go run examples.go