(The never ending story of)

# Dependency management in Go

Andrea Tosatto

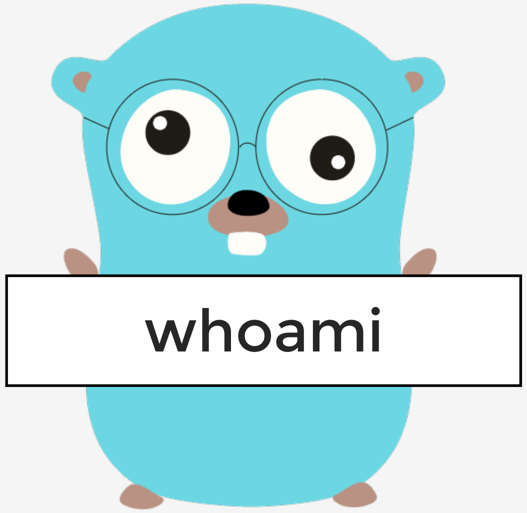
Open-Xchange

## TECH

- π-calculus, CSP, TAPL
- ❤ Golang
  Missing Golang
  Coding Golang

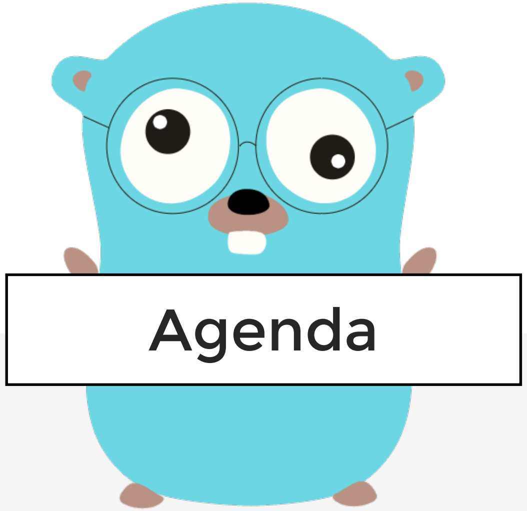
## SOCIAL

- _hilbert_
- ✉ andrea@ox.io
  andrea@tosatto.me
- atosatto



(This talk will be about)

# DEPENDENCY MANAGEMENT



# DEPENDENCY MANAGEMENT aka VENDORING

“Vendoring is the process of moving library code that your program depends on into your own project’s source”

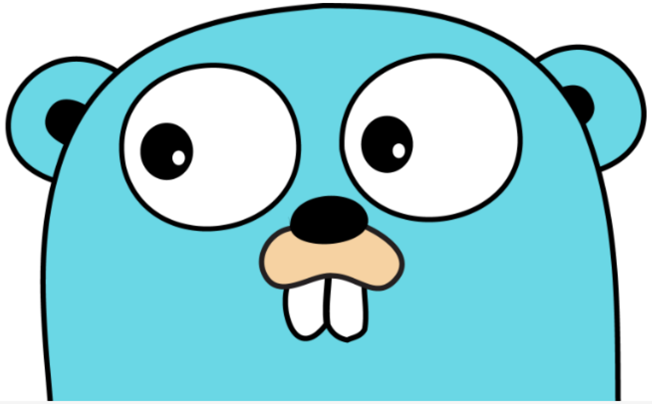
https://ukiahsmith.com/blog/understanding-go-dependency-management/

- Permit REUSE of CODE
- Allows REPRODUCIBLE BUILDS
- It’s a best practice of SOFTWARE ENGINEERING
- “If you want your developers to be happy, give them great (d.m.) tools”

(anonymous)

# AS GOLANG DEVELOPERS, WE ARE LUCKY…

- Golang ships with GREAT TOOLS
- Golang enforces STANDARDS (GOROOT, go vet, …)
- Golang has an OUTSTANDING STDLIB
- The Golang community is amazing:
  - We have GREAT DEVELOPERS building AMAZING SOFTWARE
  - The community is helping the legendary Golang designers to make the language even more awesome
- Golang has one of the cutest mascotte ever!



# …BUT…

"In Google's internal source tree, we vendor (copy) all our dependencies into our source tree and have at most one copy of any given external library. We have the equivalent of only one GOPATH and rewrite our imports to refer to our vendored copy."

*Consider an existing use of vendoring in the Go source tree: $GOROOT/src/cmd/internal currently contains copies of "rsc.io/x86/x86asm" and "rsc.io/arm/armasm" as $GOROOT/src/cmd/internal/rsc.io/x86/x86asm/ and $GOROOT/src/cmd/internal/rsc.io/arm/armasm/, respectively.  When we use those inside the Go tools, however, we import them "cmd/internal/rsc.io/x86/x86asm" and not "rsc.io/x86/x86asm".*

https://groups.google.com/forum/#!topic/golang-dev/nMWoEAG55v8%5B1-25%5D

What?!

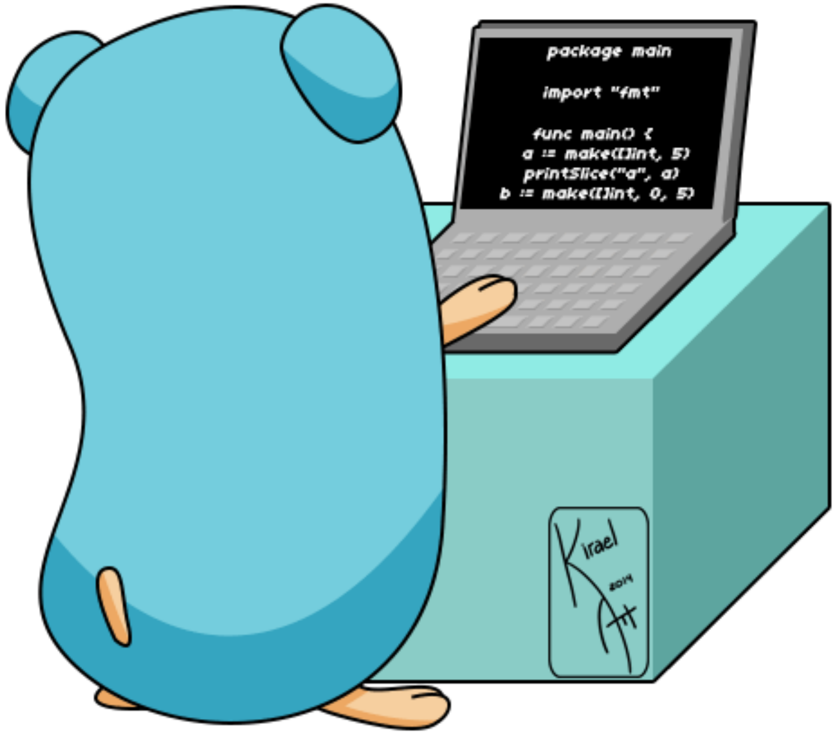
## ...SO WHAT?!

# GOOGLE USES MONOREPOS

http://cacm.acm.org/magazines/2016/7/204032-why-google-stores-billions-of-lines-of-code-in-a-single-repository/

# THE GOLANG COMMUNITY AT RESCUE

# LET’S SWITCH TO THE SHELL



```
package main

import "fmt"

func main() {
a := make([]int, 5)
printSlice("a", a)
b := make([]int, 0, 5)
```

# LINKS & RECAP

**GO GET** (go help get)
“The Golang way”

**GIT SUBMODULES** (git submodule add)
“A madness approach to vendoring for Golang programs”

**GODEP** (https://github.com/tools/godep)
“Dependency tool for go”

**GVT** (https://github.com/FiloSottile/gvt)
“The go vendoring tool for the GO15VENDOREXPERIMENT”

**GB** (https://getgb.io/)
“A project based build tool for the Go programming language.”

**GLIDE** (https://glide.sh/)
“Package Management for Go”

What about golang/dep?!

# RSC’S RESOLUTIONS FOR 2017

“I designed the Go support for downloading published packages (“goinstall”, which became “go get”) in February 2010.

[…]

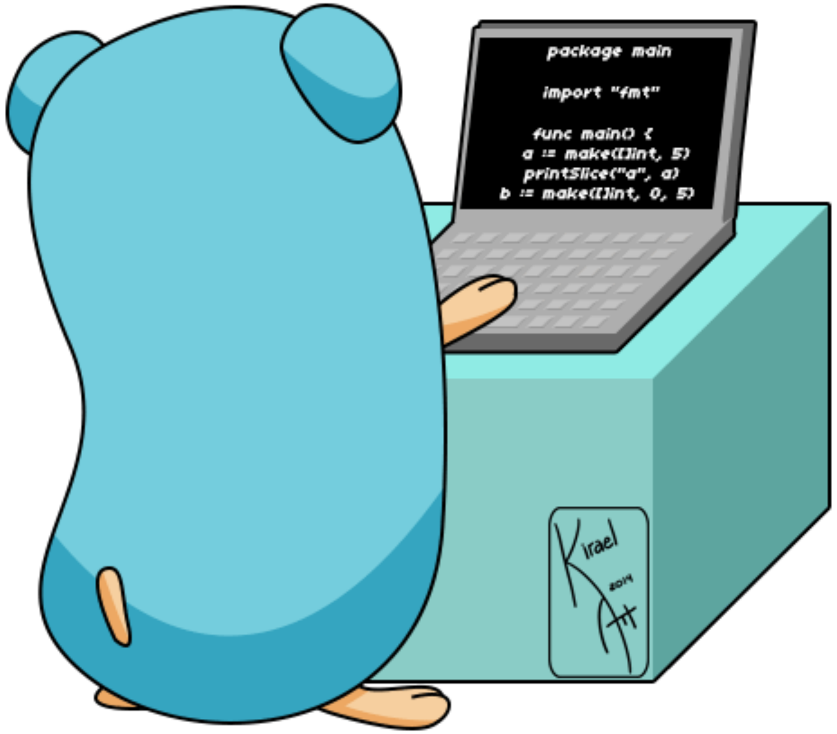
Go needs to do better here, and a group of contributors have been working on a solution. I want to make sure these ideas are integrated well into the standard Go toolchain and to make package management a reason that people love Go.”

https://research.swtch.com/go2017

# THE PROMISES OF THE GPP TEAM

- Existing projects SHOULD MIGRATE from their current tool to the official one. (We hope to make this process as automated and painless as possible)
- Projects should tag official releases according to SEMVER. Existing projects may want to retroactively tag releases. (Actually, please start doing this right away!)
- You'll stop monkeying with GOPATH, and instead work on projects with dependency code encapsulated under vendor/.
- You can commit vendor/, or not - it's up to you.

# BACK TO THE SHELL



```
package main

import "fmt"

func main() {
a := make([]int, 5)
printSlice("a", a)
b := make([]int, 0, 5)
```

# LINKS & RECAP

- golang/dep on Github
  https://github.com/golang/dep
- GPS, “your dependencies have arrived”:
  https://github.com/sdboyer/gps
- Glide’s transition to GPS:
  https://github.com/Masterminds/glide/issues/565
- The ”official d.m. tool” specifications
  https://docs.google.com/document/d/1qnmjwfMmvSCDaY4jxPmLAccaaUI5FfySNE90gB0pTKQ/
- The Go Package Management mailing-list:
  https://groups.google.com/forum/#!forum/go-package-management
- Twitter!

# Thank you!

## SOCIAL

_hilbert_

✉ andrea@ox.io

andrea@tosatto.me

atosatto