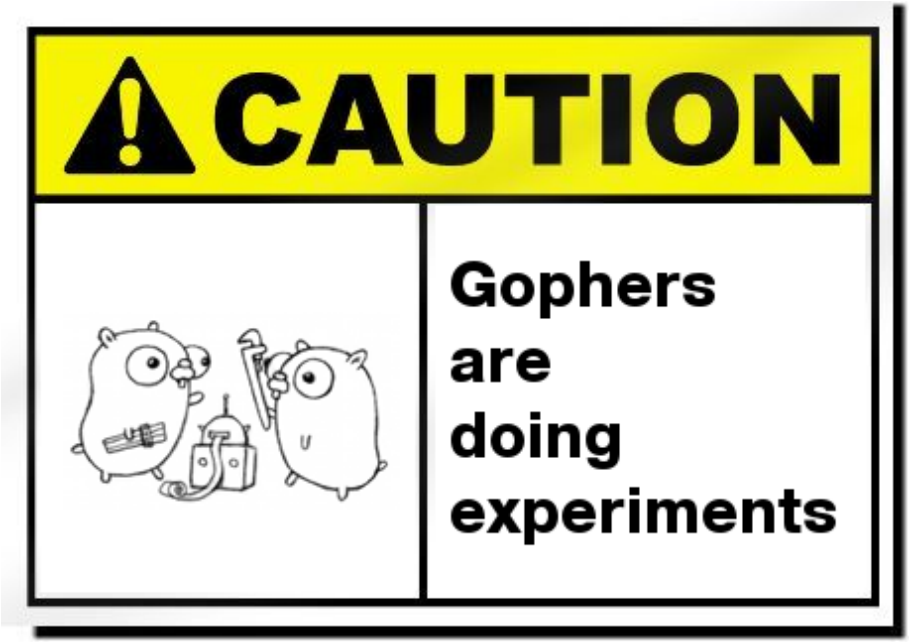
Go on Mobile

GopherCon 2015

Hana Kim
Google

Caution



The Go Mobile project is experimental. Use this at your own risk.

While we are working hard to improve it, neither Google nor the Go team can provide end-user support.

Background

Mobile support was frequently requested

Some users built their own Go binaries for Android with cgo + external linking through NDK tool chains

Some Android Apps used Go even before Go 1.4

- Camlistore android app (out-of-process model)
- Goandroid+Mandala (in-process model)
- ...

golang.org/x/mobile

Goal: Bring Go to Mobile Platforms

Why?

- Use Go to program a complete system (server/client)
- Write a single cross-platform Go library
- Bring a simple language and development tooling to mobile

Two ways of using Go

Native Apps

- Write the whole app in Go
- Use Go packages for graphics, event handling, audio, etc.

SDK Apps

- Write Android UI in Java, iOS UI in Objective-C/Swift
- Write common functionality in Go as a library

Native Apps

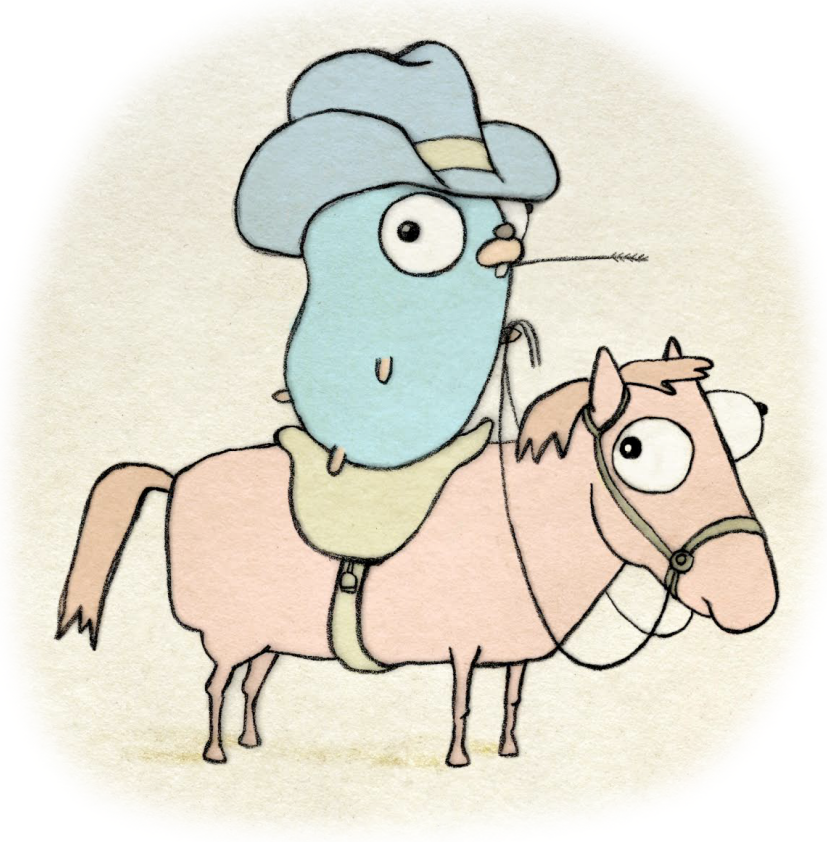
Challenge #1: Cross-platform APIs

Work for Android, iOS, and Desktop environments

Provide a rich set of APIs

Follow idiomatic Go style

Demo: Hello, Gopher!



This program uses the packages from golang.org/x/mobile repo
There is no Java or Objective-C or C in my code

What's available?

golang.org/x/mobile/...

- app: App control (https://golang.org/x/mobile/app)
- asset: Asset management (https://golang.org/x/mobile/asset)
- gl: OpenGL ES 2 (https://golang.org/x/mobile/gl)
- event: Events (https://golang.org/x/mobile/event)
- geom: Screen geometry (https://golang.org/x/mobile/geom)

golang.org/x/mobile/exp/...

- audio: Audio (https://golang.org/x/mobile/exp/audio)
- font: System font (https://golang.org/x/mobile/exp/font)
- sprite: 2-D rendering (https://golang.org/x/mobile/exp/sprite)
- sensor: Sensors (https://golang.org/x/mobile/exp/sensor)

Challenge #2: Build systems

Dealing with

- Toolchain installation
- Cross compilation for `GOOS`/`GOARCH` combos
- Android/iOS-specific build details

That is not fun!

The gomobile tool

```
$ go get golang.org/x/mobile/cmd/gomobile
```

Simplifies toolchain installation and app deployment

To install the Android/iOS compiler tool chain:

```
$ gomobile init
```

To build an Android APK and an iOS app

```
$ gomobile -target=android build
$ gomobile -target=ios     build
```

(Demo)

SDK Apps

Go as a library

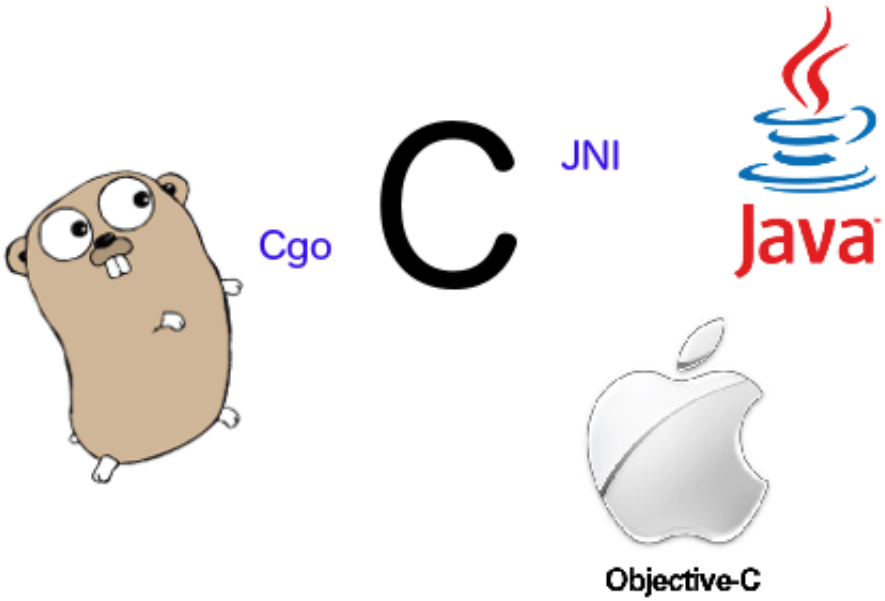
Go 1.5 can build Go programs as a library that can be used by non-Go programs

- Shared library for dynamic linking (`-buildmode=c-shared`)
- Archive file for static linking (`-buildmode=c-archive`)

Functions marked with `//export` cgo annotations are callable.

golang.org/s/execmodes (https://golang.org/s/execmodes)

Working with Foreign Languages



Manually mapping data structures and functions between languages is tedious and error-prone!

The gobind tool

```
$ go get golang.org/x/mobile/cmd/gobind
```

Automates language binding through code generation

Defines the language binding from exported Go APIs; no explicit annotation

Currently supports a subset of Go types (https://golang.org/x/mobile/cmd/gobind)

Binding Functions, Basic Types & Errors

Go API

```
package mypkg

func Hello() (string, error) { return "Gopher", nil }
```

Generated Java API

```
public abstract class Mypkg {
    public static String Hello() throws Exception { ... }
}
```

Generated Objective-C API

```
FOUNDATION_EXPORT BOOL GoMypkgHello(NSString** ret0_, NSError** error);
```

Binding Structs

```
package mypkg

type Counter struct {
    Value int64
}

func (c *Counter) Inc() {
    c.Value++
}

func NewCounter() *Counter {
    return &Counter{}
}
```

Generated Java API

```
public abstract class Mypkg {
    public static final class Counter {
        public void Inc() { ... }
        public long GetValue() { ... }
        public void SetValue(long value) { ... }
    }

    public static Counter NewCounter() { ... }
}
```

Use it from Java

```
Counter counter = NewCounter();
counter.SetValue(12345);
counter.Inc();
```

Generated Objective-C API

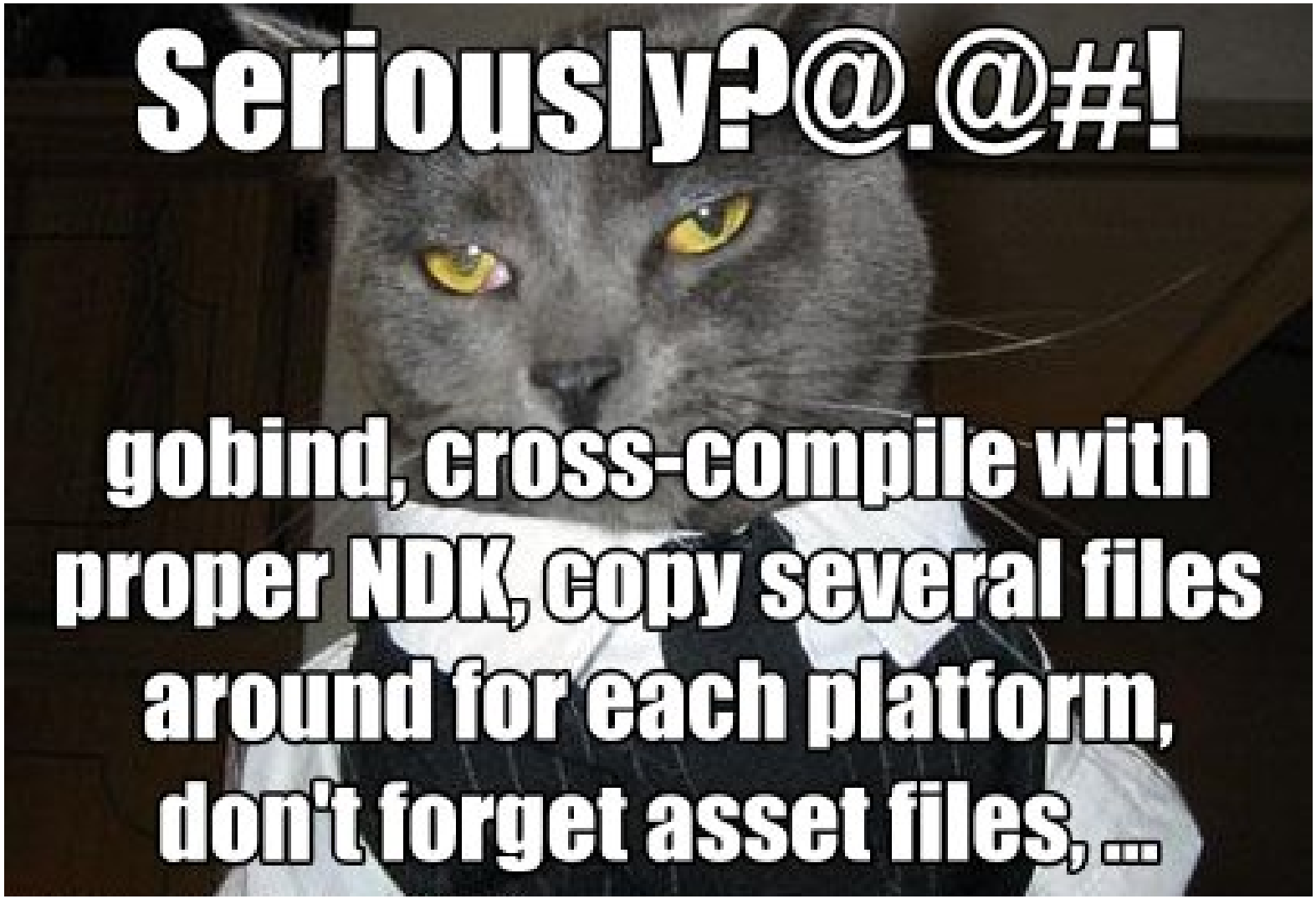
```
@interface GoMypkgCounter : NSObject { }
@property(strong, readonly) GoSeqRef *ref;
- (int64_t)Value;
- (void)setValue:(int64_t)v;
- (void)Inc;
@end

FOUNDATION_EXPORT GoMypkgCounter* GoMypkgNewCounter();
```

Use it from Objective-C

```
GoMypkgCounter* counter = GoMypkgNewCounter();
[counter setValue:12345];
[counter Inc];
```

How to build it?



The gomobile bind command

Simplifies the build process. For example, for Android,

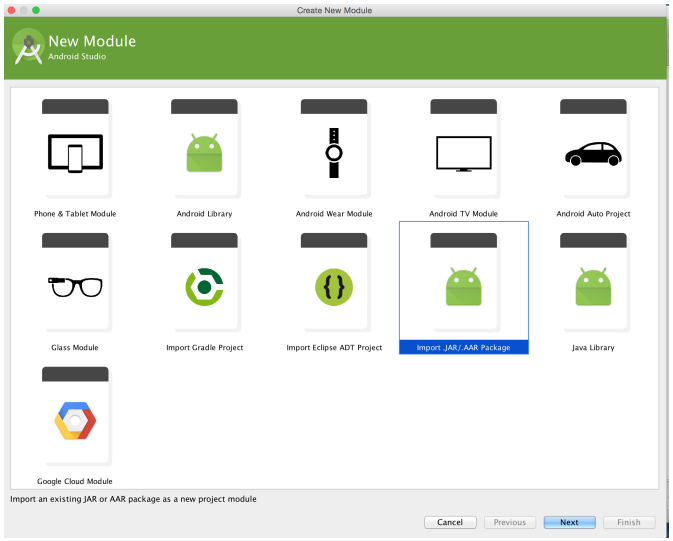
- Generates language bindings for Go packages
- Compiles Go code to a shared library
- Compiles the generated target language code
- Bundles everything into a `.aar` file (modern way to distribute android libraries)

(DEMO)

iOS support is a work in progress.

Android Studio Integration

Android Studio 1.2+ supports `.aar` import.



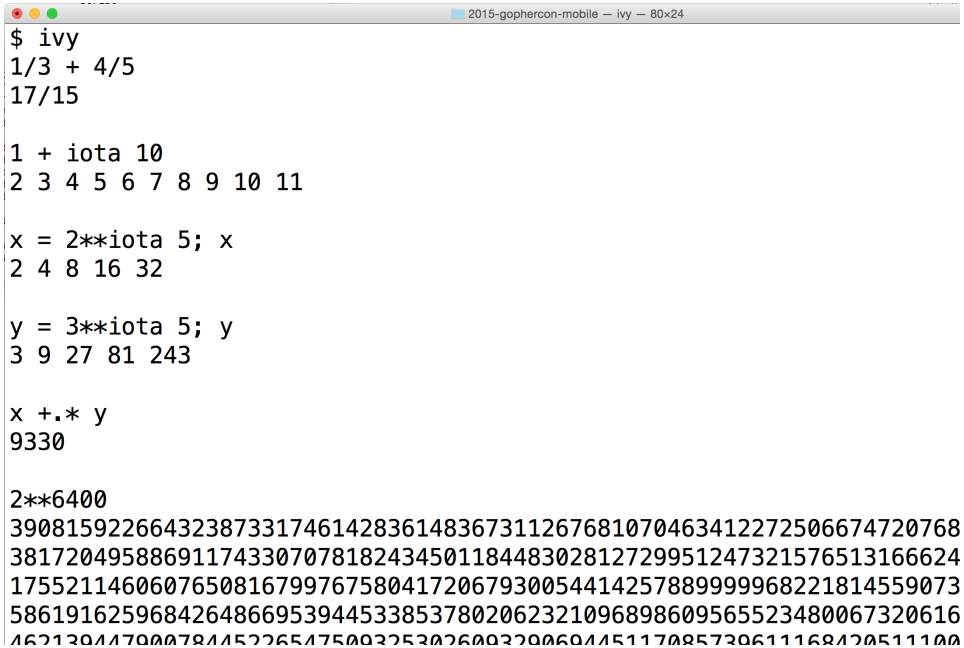
To update the .aar,

- Build script to invoke `gomobile bind`, or
- Gradle plugin to invoke `gomobile bind` and publish the output

The Story of Ivy

The Ivy (robpike.io/ivy) is a command line tool developed by Rob Pike

It's a useful desktop calculator that handles big int, rational and floating-point numbers, vectors, matrices, ...

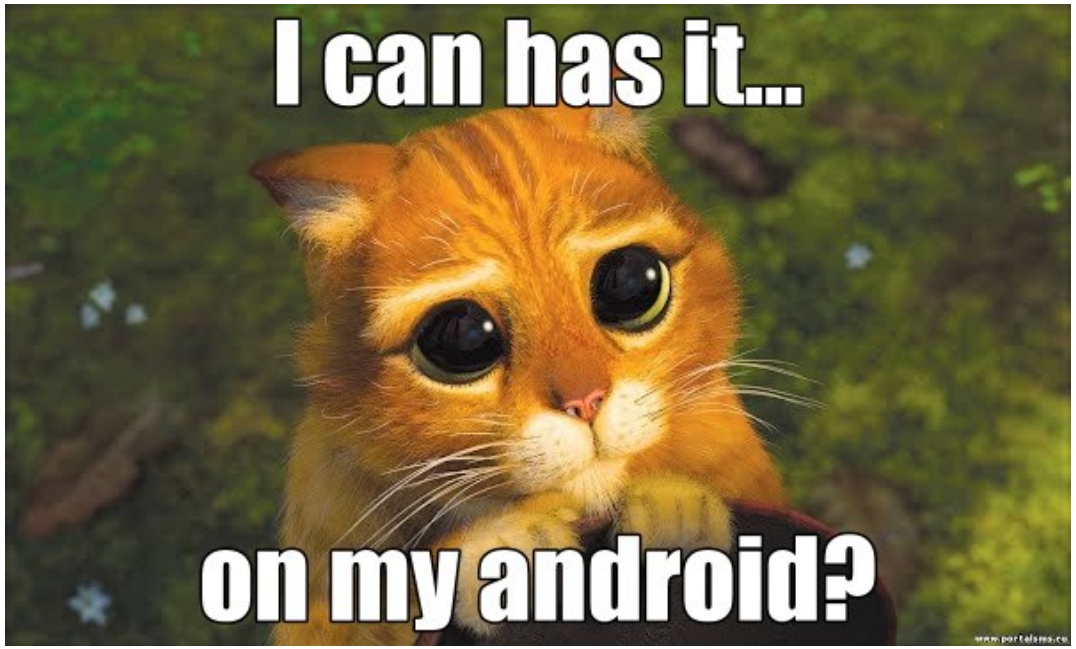


It is in fact an interpreter for an APL (https://en.wikipedia.org/wiki/APL_(programming_language))-like language

Ivy on Mobile?

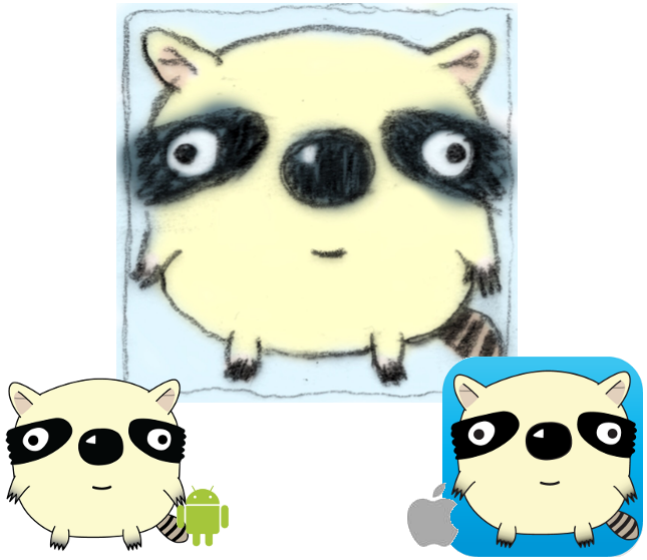
~5k lines of Go code (not including tests, docs)

Dependency on `math`, `math/big`, `math/rand`, `unicode`, ...



Rewriting in Java or Objective-C is a non-starter

Ivy apps



Ivy logo by Renée French (https://www.reneefrench.com)

Google Play Store (https://play.google.com/store/apps/details?id=org.golang.ivy)

Apple App Store (https://itunes.apple.com/us/app/ivy-big-number-calculator/id1012116478)

Gomobile bind

Write it once as a library in Go

Enjoy great language features and packages available in Go

Where are we now?

Go 1.4: Hello Android!

Released in December 2014

Can build Android apps (`arm`)

Android builder

The `gobind` tool for Java and Go language binding

Packages for cross-device apps: basic app control, OpenGL ES 2, touch

golang.org/s/go14android (https://golang.org/s/go14android)

golang.org/s/gobind (https://golang.org/s/gobind)

Go 1.5: Hello iOS!

Planned release early August 2015

Experimental support for iOS (`arm`,`arm64`)

iOS builder

Go 1.5: Go programs as libraries

Can call Go functions from foreign language in a clean way

golang.org/s/execmodes (https://golang.org/s/execmodes)

Go 1.5: Better tools & more packages

`golang.org/x/mobile` repo getting better

- The `gomobile` tool for mobile app/library build
- Extended `gobind` tool: Objective-C binding
- `golang.org/x/mobile/exp`: experimenting with audio, sensor, sprite, ...

Go 1.6+

- Improvement in GL/UI packages
- More APIs available to "pure Go" apps
- Testing, profiling, debugging
- Support for more platforms (e.g. android/x86, iOS simulator)
- Richer type support in `gobind`
- IDE integration

Contributions from Go community

git log | word_cloud

Thank you

GopherCon 2015

Hana Kim
Google