Modules, the Athens project and beyond

27 June 2019
Speechmatics office

What are modules

- Collection of Go packages
- Have a version and use semantic versioning
- Specify which code is included in a build
- See https://roberto.selbach.ca/intro-to-go-modules/ for a great worked example

go.mod

- File in the root of a module specifying:
  - Module name
  - Modules included in the build
- Can be updated by hand or by tools

go.mod

- Create a module using:

```
go mod init example.com/newmodule
```

- This can read existing godep, glide, dep and other dependency management systems files

go.mod example

```
module example.com/newmodule

require (
	golang.org/x/text v0.3.0
	gopkg.in/yaml.v2 v2.1.0
)
```

The go.mod file of Athens

Adding modules

```
go build . # Automatically adds missing modules

go get example.com/anothermod # Add module

go get -v example.com/anothermod  # Verbose

go get -v -x example.com/anothermod  # More verbose, also works for go build

go get -u   # Update all dependencies. Will change in 1.13 (https://go-review.googlesource.com/c/go/+/174099)
```

Useful commands

go list -m # print path of main module

go list -m all # get all modules used

go mod tidy # adds missing and removes unneeded modules

go clean —modcache # Remove entire download cache

go build —mod readonly # Build is not allowed to update go.mod, use this in CI

go.sum

- File in the root of a module specifying the checksums of the modules

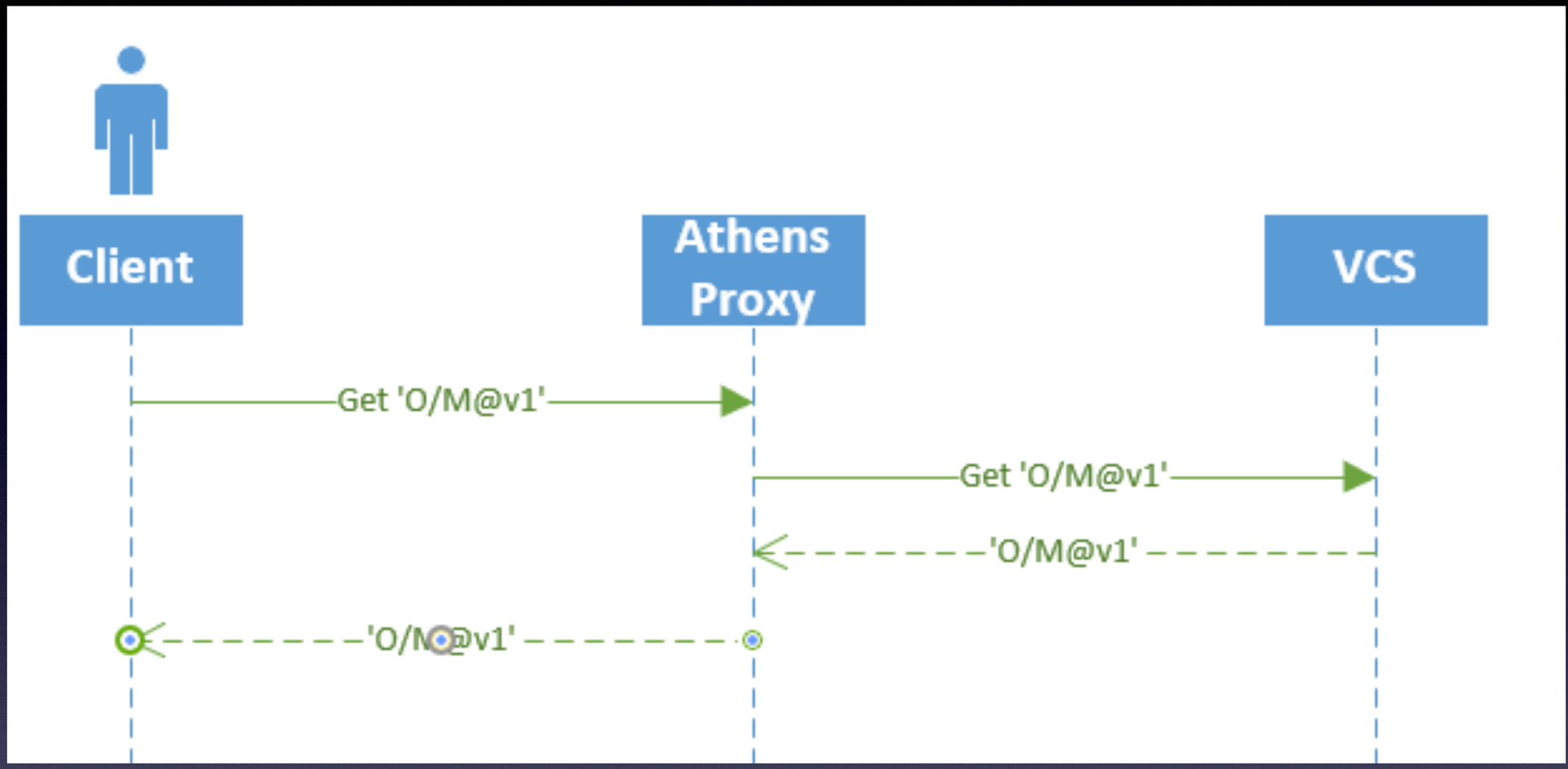
- This makes sure tomorrow's build uses the same code as today's

- Should be in version control (same for go.mod)

The go.sum file of Athens

Project Athens

- A Go module proxy that stores copies of modules so:
  - download performance is better
  - local copies of modules are available
  - custom logic can be added (eg access control)
- Microsoft started this project



Behaviour when nothing is cached
© Athens website

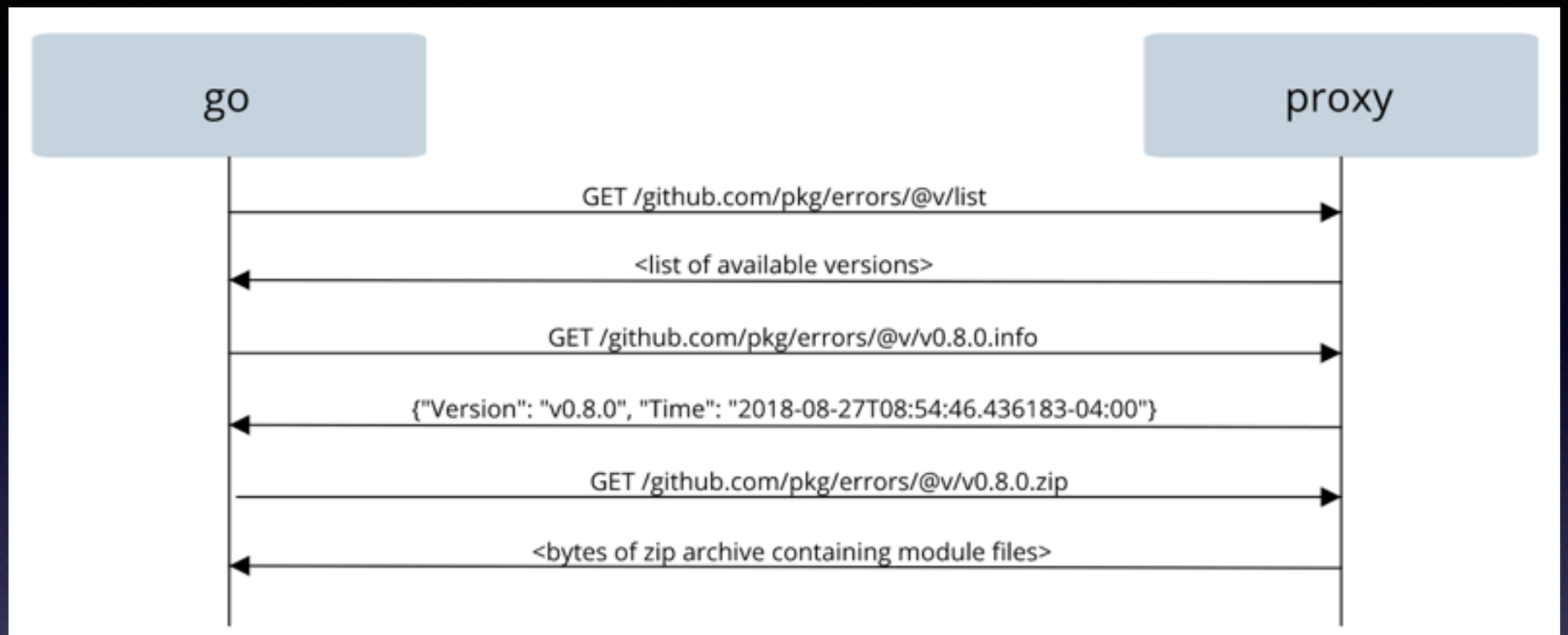
Behaviour when cached
© Athens website

GOPROXY (1.11)

- Based on the GOPROXY env variable which can have the following values:
  - “off”: does not download anything
  - not set, empty string, “direct”: always go to Version Control System (VCS)
  - URL: use the proxy server specified
  - Can also use file:///filesystem/path

Download protocol

- GET $GOPROXY/<module>/@v/list
  - returns a list of all known versions of the given module, one per line.
- GET $GOPROXY/<module>/@v/<version>.info
  - returns JSON-formatted metadata about that version of the given module.
- GET $GOPROXY/<module>/@v/<version>.mod
  - returns the go.mod file for that version of the given module.
- GET $GOPROXY/<module>/@v/<version>.zip
  - returns the zip archive for that version of the given module.



Getting github.com/pkg/errors
© https://roberto.selbach.ca/go-proxies/

Let’s do a test

- Git clone athens (master branch)
- Clear the module cache between runs: go clean —modcache
- Run go mod download with or without Athens:
  - direct
  - local Athens instance

Overriding caching in Athens

- Athens can be setup to do the following for a particular module:
  - Go directly to an upstream proxy (D)
  - Exclude it and the submodules (-)
  - Include it (+)

Example config

```
# This is a comment

- github.com/azure

+ github.com/azure/azure-sdk-for-go

# get golang tools directly

D golang.org/x/tools
```

Additional features

- Upstream proxies: always forward a request to a proxy
- Filter on version: approved list
- Proxy a Checksum DB

Running Athens

- Docker command:

```
export ATHENS_STORAGE=~/athens-storage
mkdir -p $ATHENS_STORAGE
docker run -d -v $ATHENS_STORAGE:/var/lib/athens \
-e ATHENS_DISK_STORAGE_ROOT=/var/lib/athens \
-e ATHENS_STORAGE_TYPE=disk \
--name athens-proxy \
--restart always \
-p 3000:3000 \
gomods/athens:latest
```

- Can be installed in Kubernetes as well: see https://docs.gomods.io/install/install-on-kubernetes/

Storage options

- Athens supports a lot of storage options:
  - Memory
  - Disk
  - Mongo
  - Google Cloud Storage
  - AWS S3/Minio

Problems with GOPROXY/ Athens

- All or nothing: all the packages have to be gotten from the proxy or it will fail

GOPROXY (1.13)

- In Go1.13 GOPROXY env variable is a comma-separated list of proxies
  - If a proxy returns 404 or 410 for a module the following proxy in the list is used
  - GOPROXY=https://proxy.example.com,direct
- GONOPROXY: avoid the proxy for modules matching a pattern, eg:
  - GONOPROXY=*.corp.google.com,rsc.io/private

Go Checksum Database

- Hosted database of checksums for module versions
- Google provides one at https://sum.golang.org, companies can run their own
- GOSUMDB sets the checksum database
- GONOSUMBD is a list of module path prefixes that don’t use the checksum database (overrides GOPRIVATE)
- Enables new use cases: eg tells you new versions of a module are available (https://index.golang.org/index)

GOPRIVATE

- GOPRIVATE is a list of module path prefixes for which neither proxy or checksum database should be used

- Example: GOPRIVATE=*.corp.example.com,rsc.io/private

- GONOSUMBD and GONOPROXY override GOPRIVATE

Go module mirror

- Google are (already) hosting a module mirror at https://proxy.golang.org
- In 1.13 the default value of GOPROXY will be “https://proxy.golang.org,direct”
- This means it tries to download modules from the global Go module mirror first before trying to contact the VCS
- If you don’t want to leak your internal server names, set GOPRIVATE
- The Go module mirror has a privacy policy

Let’s do a test (again)

- Git clone athens (master branch)
- Clear the module cache between runs: go clean —modcache
- Run go mod download with various GOPROXY options:
  - direct
  - https://proxy.golang.org
  - local Athens instance

Performance

	Direct	proxy.golang	With Athens
First	1m 20s	0m 43s	2m 32s
Second	1m 28s	0m 49s	0m 12s
Third	1m 25s	0m 41s	0m 12s
Fourth	1m 15s	0m 20s	0m 11s

Alternatives

- JFrog Artifactory can work as a module proxy as well
- You have to use the JFrog CLI, eg:

  jfrog rt go-publish go --self=false --deps=ALL

- It has enterprise features such as access controls

GoCenter

- JFrog also provides GoCenter (https://gocenter.io) for public modules
- Their goc binary is a wrapper around go which tries GoCenter for modules first and if not found tries the normal way

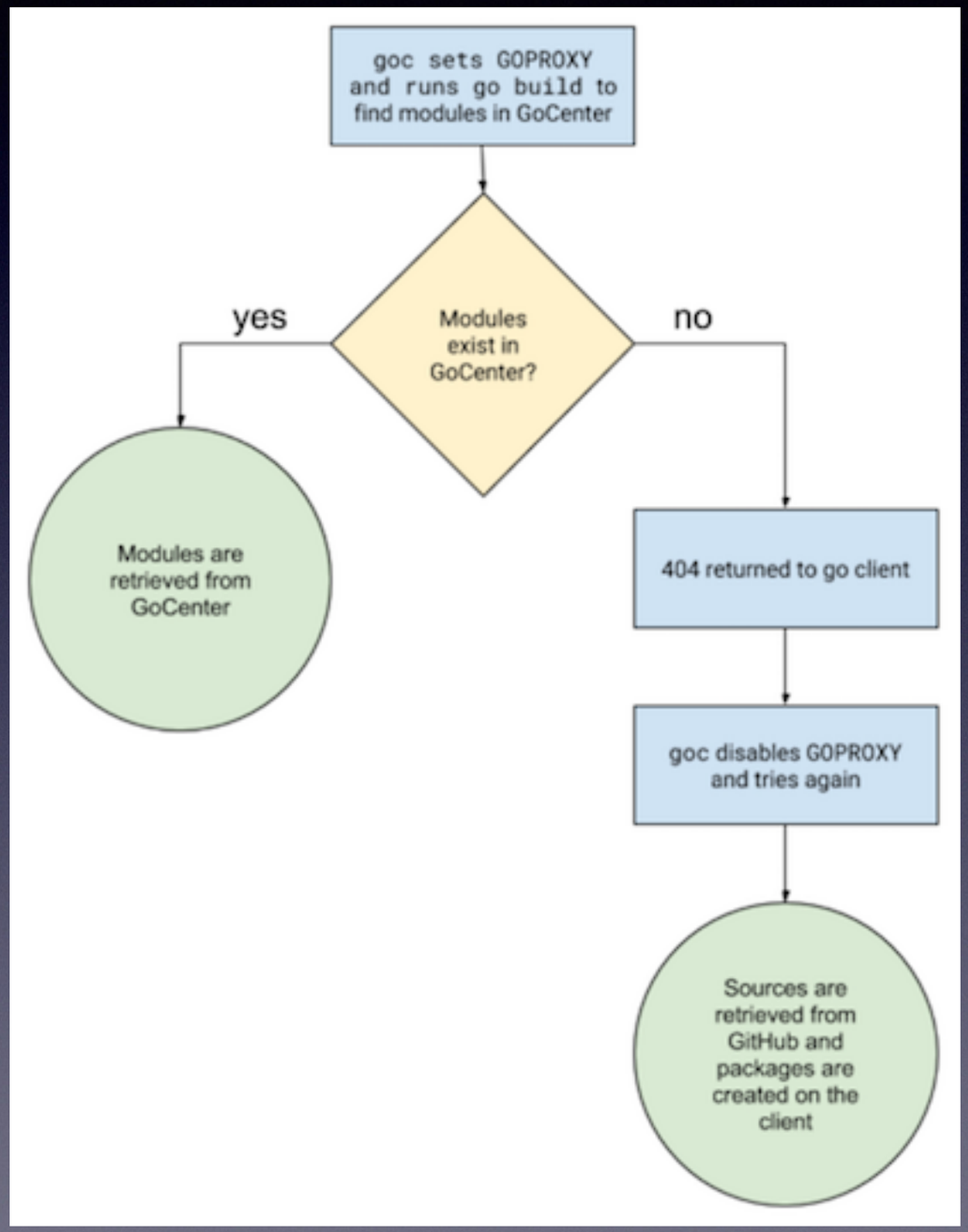


Image © JFrog

Conclusions

- https://proxy.golang.org will be a default in 1.13 and it's quick: this is a win for everyone
- The 'direct' fallback makes sure you can always get to your modules
- Use GOPRIVATE (or GONOPROXY/ GONOSUMDB)
- Run a local Athens instance for your private modules to speed up your local and CI builds

Thank you.

Any questions?