# Go Working Group – 3/20/2018

Attendees: Drasko & Janko (Mainflux), Fede (Cavium), Trevor, Tingyu & Jim (Dell), Steve (IoTech), Tony (Canonical), Alberto (NOV)

Discussion and action items as a result of meeting in RED.

**Last Go Lang Project meeting! Per on-line vote, this work will move into the Core Working Group.**

**Look for new Snappy Project meeting coming to Tuesday time slot – time moved to 1pm CDT.**

## Old business

*Mono Repo*

- Docker builds (not working yet)
  - Dockerfiles in branch are working – Fede to submit PR to move to master.
  - CI build/tests working on ARM are needed to complete everything appropriately
  - Working on ARM 64, best to keep in from preventing PR merge until we get a few issues ( check with Jeremy)
  - Tony – Snap using mono repos almost there – some files changed location and need to be addressed.
- ARM32 build
  - Is there a MongoDB binary for Arm32? Steve needs for Raspberry Pi.
  - To be put on to-do list after Arm64 all working.
  - We may need to consider getting and building our own infrastructure for these (Mongo, Consul, etc.)
- Consul integration
  - Trevor providing great working example via core services. Will export out to other services once work is completed.
  - Still need to demonstrate with YAML
  - Still need to update config-seed
  - Tony - Is registration or config-seed come first when you have consul on?
  - Registration should be last step in order to avoid sleeps.
  - Dell team looking at and will get back to Tony. May need to relook DS requirements.
- README consistency
  - Action: Jim to look at and try to improve once the services, Makefile, etc. are all a little more stable.
- Glide lock (and .gitignore)
  - Action: We should put versions in the Glide file so that new pull or CI build all get and use the expected version of a dependency.
  - The Glide Lock therefore can be added to .gitignore (and prevent someone from getting and using someone else's or getting one before a make install is done.
  - Drasko and Mainflux tried GoDep (now entered stable production). Works similar to Glide. Revisit post-California

*0MQ alternatives*

| Product | Native Language Bindings, OS support, etc. | Brokerless | License | Used By |
|---|---|---|---|---|
| Thrift | C, C++, Go, Python, Java, | Yes | Apache 2 | Cloudera, Evernote, Facebook, Siemens |
| Nats | Go, Node, Ruby, Java, C, C# | No - server | MIT | Baidu, Siemens, HTC, Pivotal, VMWare, |
| gRPC | C, Go, Java, Python, Node | Yes | Apache 2 | Square, Netflix, Cisco, CoreOS |
| Nanomsg | C, Go, Java, Python, Rust, Node | Yes | MIT | ??? |

*Intel DPS:*

Steve: Intel DPS – pub/sub and point to point. Anyone look at? Needs research. A bit experimental and in C?

- Probably not a solution at this time.

*Thrift:*

- Apache Thrift allows you to define data types and service interfaces in a simple definition file. Taking that file as input, the compiler generates code to be used to easily build **RPC** clients and servers that communicate seamlessly across programming languages. Instead of writing a load of boilerplate code to serialize and transport your objects and invoke remote methods, you can get right down to business.
- Thrift does require a "compiler" to create/generate a thrift file to use by the implementations.

*Nats:*

- NATS **Server** is a simple, high performance open source messaging system for cloud native applications, IoT messaging, and microservices architectures.
- Server binary is available for Linux (x86, x86_64, ARM), Windows (x86, x86_64), and macOS. Server image is available in Docker image. Server is 2.4 MB in size
- Zero Quality of service
- Nats Streaming – wrapper around Nats with QoS (Apache 2); may have some limitations like orphan connections

*gRPC:*

- A high performance, open-source universal **RPC** framework.
- It is a Cloud Native Computing Foundation project.
- Also requires a compiler for protocol buffers.
- Pub-sub over gRPC (but a hosted solution, we could not leverage)

Nanomsg:

- A socket library that provides several common communication patterns. It aims to make the networking layer fast, scalable, and easy to use. Implemented in C, it works on a wide range of operating systems with no further dependencies.
- There is now also an implementation of nanomsg in pure Go: mangos.

- http://sealedabstract.com/rants/nanomsg-postmortem-and-other-stories/

### Could we use ZeroMQ for now?

- We need Interfaces to allow for easier replacement
- Lack of libraries in Window for Window's developer and issue and difficulties on ARM.
- Zero MQ does work fine on Alpine Linux
- Compilation natively is the issue with gRPC too
- MQTT is something we need to support long term.

Consensus agreement that for now, we stick with ZeroMQ through California release. Dell team will look at interfacing the message elements such that 0MQ could more easily be replaced by a user.

Longer term, we do need to address the needs of the Windows developer and provide more ease of use for ARM. We'll relook at these and potentially offer alternatives post California release.

The community is encouraged to do some POC work with alternative technologies like those listed above to be able to provide more information for the relook after California.

## New business

*Any Other Open Items??*

- Tony – OpenIoT Summit: overview of Mainflux. They thought Go was a bad choice per performance of encryption.
- Is an issue: Go assembler for optimization for certain operations. Much for Intel; little for ARM. Also work going on for ARM to bring it up to Intel standards.
- Fede – in Go 1.10, they have already changed crypo to assembler.
- Drasko – from Intel side, a lot of work with Crypto for digital ledger projects in Go.
- Tony – definitely a problem that is real. We should track.
- Steve – Alpine uses 1.9 (building in the container)