

Go Working Group – 2/20/2018

Attendees: Fede & Chencho (Cavium), Drasko, Darko & Janko (Mainflux), Tony (Canonical), Alberto (NOV), Trevor & Jim (Dell)

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

Old business

- Are we there yet? What else needs to be accomplished before creating the mono repo?
 - Technically, we are there. No additional files, scripts, etc. are believed to be needed
- Discussion
 - There was much lively discussion as to the advantages and challenges of a mono repo with a single version versus having separate repositories (with their own version) for various parts of the architecture (like core, device services, application, etc.). The general problem is one of convenience and simplicity of one repo for builds, etc. versus the possibility that a shared library, like core-domain, would need to be altered and then cause all the users of that particular library to upgrade to the next version of everything to get the fix. In other words, the mono repo/single version does not provide as the ability to get more discrete elements of EdgeX.
 - While there were some general reservations and doubts about the long term effectiveness of the mono repo and, in particular, a single version for all the code in a mono repo, the decision was made to move ahead with the mono repo creation and re-evaluate in one to two months. That still allows some time before the California release to readjust the repo organization and versioning if necessary.
 - Decisions:
 - Create the mono repo in the EdgeX Github (edgex-go) on Feb 26th. Fede's script will pull all the latest code from the other go repos at that time. A PR is already setup and just needs to be approved/accepted. Jim will coordinate with Fede and Jeremy and approve the PR on the 26th.
 - The old Go repositories and the new edgex-go repository will both be used, and both will have to be maintained by Go developers until March 2nd. At which time, unless there is complaints or other issues, the old Go repositories will be archived. Jim will coordinate with Jeremy on the 2nd to archive the old repo.
 - In April, Jim will schedule time to review the current Go efforts and survey developers for changes/alterations they'd like to make to the structure.
 - All maintainers/committers from the existing Go repositories will become maintainers and committers of the new mono repo. WG chairs are encouraged to maintain the acceptance/approval of PRs based on areas of responsibilities (i.e. Janko and his committers/maintainers manage code related to applications WG, Jim and his committers/maintainers manage code related to core and supporting services, etc.).
 - The Java and Docker containers (like volume, mongo) will continue to be maintained as separate repo.

New business

OMQ vs Nats, gRPC

There are issues with use of ZeroMQ – especially as it relates to cross compiling and Windows development. Drasko thinks some of this might be solved with dynamic compiling. Jim reports that the OMQ client library used may not compile on Windows based on some info from the forums.

Would it be easier/better to solve the OMQ compiling difficulties and issues or switch to another message bus? In general, it would be good to have a native Go lightweight, message infrastructure to use between core data and distro (and rules engine). There are many options to include Nats, gRPC, etc.

Action: Jim will schedule more time to discuss during future meetings.

Meeting time and continuance?

- Time of this meeting still good – every other Tuesday 2pm CST?
- Fold back into Core with completion of mono repo?

Unanimously, the group thought continuing a separate Go project and meeting was necessary at least until the mono repo is in place. The 2pm CST works well for all – including members in Europe.

Other Open Items

Tony brought up getting license files moved to the docker containers. This topic is generic and Jim is moving it to a topic for the Core Working Group on Thursday.