

Architect's meeting, Jun 30, 1pm CDT

Extra meeting for June due to # of architectural issues.

Attendance:



Some attendees may have joined after the call started when this record was captured.

Topics

Old

- Complete our look at how to review 3rd party libs/modules - get a read out from James and Tony on the evaluation criteria. <https://github.com/jamesgregg/1947-explore>
 - Thanks to James, proposal now in the Wiki: <https://wiki.edgexfoundry.org/pages/viewpage.action?pageId=46760301>
 - Per last meeting:
 - Tony and James are going to get together and, using James's list of criteria from his 1947 explorer, narrow the list down to the important/most critical criteria
 - Jim volunteered to use that criteria to explore the 30 or so packages that have no version number (have a dynamically generated version tag) currently associated with edgex-go as a test of the criteria and report back to the group.
 - We'll formalize decision around criteria and how it is applied after that.
 - Criteria established by Tony and James:
 - **Total increase in new imports:** does the new import introduce additional import dependencies, if so, how many?
 - Ensure that **every** one of the new dependencies is checked for the same criteria.
 - **Releases/Tags:** count
 - We should avoid new imports that have never had a release and/or tag. How many is too few, this is a judgement call and probably involves also considering how long ago the last release was, and how far apart releases have been done.

- **Contributors:** count
- **License** - what is the license, and is it Apache 2.0 compatible?
- **Stars/Forks/Watchers:** counts
 - These are all indications of how wide-spread the package is used.
- **godoc.org metrics:** count
 - The individual godoc pages hosted by godocs.org include metrics at the base of the page which indicate how many packages import the package
- **Subjective opinion of the reviewers** – at the end of the day, we rely on our reviewers to vet new code. Reviewers should give thought to whether the code is improving our project, whether we'd be better off to implement the functionality ourselves, and at the same time considering whether this new import itself comes with too many dependencies (e.g. go-kit).
- Our overall goal is to keep our dependencies low!! – it is an issue today.
- Jim to start the review of the 30 packages that have dynamically versioned tags using the criteria above and reporting back to this group.
- Current PR that this impacts: Go Diff lib – used in unit testing; ok for this PR. Flag that caused this to be a concern was “no longer being maintained” in the lib’s readme.
- Cloud and team still working the issues with this PR, but library is approved for this PR. <https://github.com/edgexfoundry/go-mod-core-contracts/pull/236>
- We should also avoid having different versions of a package (like this one) in go.sum.
 - Does Go pick the latest of a package when multiple is pulled into go.sum? Mike to research.

New

- Does EdgeX work with IPv6? If not, what work needs to be accomplished and what are the impacts to the code base?
 - Cloud and team did successfully run EdgeX 1.2.1 in Docker and ran the blackbox tests without issue under IPv6. However, Docker Compose/Docker network is still using IPv4.
 - Action items:
 - We should better understand what does/doesn’t work; we have some hidden gotchas (like default 0.0.0.0 addressing).
 - Only place it impacts is where we listen/serve – and configuration (like Kong).
 - Could we do a native deployment to test? Does Snap do that and would that work to test this?
 - Decision
 - Release-note what we have tested to date for IPv6 (caveat where there may be issues) for Hanoi
 - Take this issue on for Ireland
 - Focus on external looking elements (Kong or device service separated from rest of EdgeX) and not focus on overlay network (service to service on the Docker network)
- PR Template for conventional commits is now in place for all repositories for all PRs but without TSC approval. It doesn’t appear to be affecting any problem. We need to finalize the shape of this and officially approve the template by the TSC.

- Tony provided a slide deck making an argument for returning to documented commit oneline summaries and commit messages (see deck in the Wiki).
- Others (majority) still believe there many benefits to use of conventional commits and the current PR template (see <https://www.conventionalcommits.org/en/v1.0.0/>)
- Mike/Tony and Lisa to explore the current template, how we use it and conventional commit policy and provide TSC a recommendation about how to proceed going forward.
 - Making domain check off a required field on the commit template may be one such tweak to the existing system.
- It is likely that we will start to use a bot to enforce what is provided in the template (this was looked at in May – explorer work there was already done and may just need to be refreshed).