Architect’s meeting, April 16, 7am-10am

Attendance:

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

- Config override – env var names; case sensitivity
  - https://github.com/edgexfoundry/go-mod-bootstrap/issues/69
  - We want all UPPER CASE going forward
  - To address non-backward compatible changes for now, we allow for UP CASE or Pascal/CamelCase. We drop Pascal/CamelCase at the next major release point.
  - Lenny has PR out there. Just needs approval.
  - Provision Watcher – the device profile is embedded in the provision watch making the PW unnecessarily large. Going forward, we wat the tight coupling to be removed so as to reduce the size of the PW. However, due to the late hour in implementing this for Geneva, it was decided to punt this refactor until Hanoi
    - There is a similar issue in that devices also have the embedded Device Profile. This too may be taken care of in Hanoi
    - https://github.com/edgexfoundry/go-mod-core-contracts/issues/226
    - Jim to check with Cloud to make sure delaying this until Hanoi is not going to be an issue.
- Apache 2 licensing requirements (per Tony E message and Brett P (LF) and Ducan (IOTech) responses.
  - Attribution.txt – is part of all source repositories. It lists 3rd party modules, code included with EdgeX
  - License states that “Derivative work” (docker images, snaps etc.) should include this file (long with LICENSE file containing the Apache 2 license). Today the code base includes it, but the artifacts to not.
  - Everyone was ok with adding the Attribution.txt file to all artifacts at the root (where the binaries are located).
  - There will be a best effort made to try to get this in for Geneva – but make it required and enforced for Hanoi.
  - Jim to coordinate with WG leads to get issues in for Hanoi/Geneva for all services.

- Address policy on patch/minor release based on discovery of CVE or other security issue (high/critical) - would we force dot release of all services, just affected ones, ??? For Consul and Mongo, we just minor rev'ed those services and containers. What if issue was in one of the go services? Would we have to minor release all (example go from 1.2.0 to 1.2.1) on all services even if only one was affected?
    - Dot release of 3rd party service alone is no brainer. When it involves one of our releases, different story.
    - Would this technically be possible to have a dot release of one (like core-data) and not the others?
    - Set up policy, based on severity of the issue, on when a dot release is scheduled to be released. Avoiding a release after release on short term cycle.
        - Critical – fastest path to get out a patch (on demand)
        - High and below – addressed with the next release
        - We should track High's more acutely
    - Decision is no real change to policy. We should try to track the number of issues a little more closely (especially Highs) to see if we should be doing something more later.
- Review Hanoi Planning meeting topics (architecture, business, scope, unconference)