Core Working Group Agenda 05/24/2018 meeting

Attended by: Keith & Andy & Steve, Tony & Ian, Rodney & Brad, Eric & Jim, Dan Bryan, Jeremy, Janko. Others may have joined the call after the roll was taken at the beginning.

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

Old Business

DevOps items
- Jeremy – any issues/updates we need to discuss
- Same action items for TSC yesterday – wrap up for California
- Dependency manager discussion – remove Glide and go to GoDeps

Documentation Hosting
- Still working; host on LF infrastructure, need build process to deposit in appropriate place
- Andy – status of
  - training and guidance for new resources in the process of doc generation
  - record a training session and post to Tech Talks
  - Action Item for Andy – look at scheduling a Zoom session and record that for reference; looking at next two weeks. Any day after Monday – working with Brett and record and make available on Wiki.
- Jeremy working on building docs on LF hardware and publishing

DS Requirements Discussion
- New DS SDK meeting on Tue’s. Connect with Brett or Tony to get involved.
- Plan, schedule, and issues to be resolved in upcoming meeting and finalized by Jun 4 meeting
- Changes to requirements on floats (float base 64) and response objects – see docs
- Next DS meeting is next Friday (June 1)
- Tony locking down agenda for SDK discussion day – Jun 4

Performance Testing
- Keith – provide information around possible IoTech performance tests contribution by CA F2F meeting
- Hope to make some of their work as it relates to performance with POC available to community soon.

EdgeX UI
- New project group under Core WG. First meeting Thur May 24th @6am PDT
- Next meeting in 3-4 weeks will outline roadmap and direction for Delhi

Palo Alto F2F
- Please sign up (connect with @BPreston) if you are going to be there (in person or on line)
- Please visit
The document contains information about a meeting agenda and topics. It includes:

- Review of Delhi scoping
  - Early disagreement with proposed in/out scoping?
  - Anyone wants to advocate adding/removing something
- Review of Architect Tech Debt
  - Early disagreement with proposed topics
  - Calls for additions/deletions from that list
- Review of Architect Enhancements
  - Early disagreement with proposed topics
  - Calls for additions/deletions from that list
- Agenda will be considered set by mid next week! There is little time for extra subjects to be introduced at Palo Alto without prior arrangement.

**TSC Working Group Chairs**
- Each working group elects its own chairperson. That person serves on TSC.
- WG voter lists to be reviewed here.
- Current nominations under core (need to have this completed and voted on by 4th!!!)
  - Core WG Chair: Trevor Conn (Dell)
  - Test QA WG Chair: Andy Foster (IoTech)
  - Applications WG Chair: Janko Isidorovic (Mainflux)
  - DevOps WG Chair: Jeremy Phelps (LF)
  - DS/SDK WG Chair: Steve Osselton (IoTech)
- Other working groups to nominate via their meetings
  - Security
  - System Management
  - Vertical Solutions
- 3 at-large to be elected on June 15 (Jun 1-8 nominations)
- TSC Chair to be elected from WG Chairs & TSC at-large – June 22

**New Business**
**Code PR/Review/Acceptance Process for Post-California**
- Proposal to follow more of the Docker model for code review/acceptance submitted by Drasko
- [https://github.com/moby/moby/blob/master/project/REVIEWING.md#code-review---status2-code-review](https://github.com/moby/moby/blob/master/project/REVIEWING.md#code-review---status2-code-review)

"Changes to code must be reviewed and approved (LGTM'd) by a minimum of two code maintainers. When the author of a PR is a maintainer, he still needs the approval of two other maintainers."

**Thoughts and Questions**
- If two reviewers LGTM, can the original creator of the PR approve and merge himself? Or is it up to the second reviewer to Approve/Merge?
- If two reviewers LGTM and the PR has to be re-merged due some other changes to the trunk which have been merged in the interim, does the review process need to happen all over again? (I would advocate for "no")

**Discussion**
- It’s not just the docker model – the plus 2. Good idea to have domain experts review to approve.
o Should we require issue/PRs to be filed before writing code. But then working groups are going to have to review PRs (extra work load).
o GitHub has tooling to help out in this area. Use folder code owners to require specific people to sign off on it. Get some granular approval there.
o Other open source projects – committers and maintainers and it takes majority of committers to +1 it.
o Can look at putting a min number on committers.
o Committers can also come in and vote -1.
o Need to make sure that “rails” don’t get in people’s way.
o Eventually – interface review board?
o Working groups may not be aware of patterns/work in other working groups. May cause divergence in APIs
o Should take a stab at our own improved processes around PRs (versus adapting/using another projects rules)
o Scheduling is a bit of an issue as well.
• Action: everyone to consider and have ideas for how we address at next week’s meeting.
• Action: goal is to assign a person or two to come up with new process/steps after next week meeting for review and adoption for post-California

New Export Distro design for Delhi: based on SDK
• Export Distribution is essentially an EAI (enterprise application integration) problem
• Export Distribution is about taking incoming Events/Readings and:
  o Validating the data in the Event/Reading (valid device, valid readings, etc.)
  o Filtering (per device or value descriptor today, but that could be open to other filters like temps between a range, etc. later)
  o Transforming the data to the preferred format (XML, JSON today but could be YAML, TOML, CSV, etc.)
  o Perform other transformation operations (optionally compressing it, optionally encrypting it today)
  o Delivering the data to the endpoint of choice (which is MQTT, MQTTS, HTTP, HTTPS, cloud, Rules Engine, analytics provider etc.).
• Going forward, having one export service do this is not scalable. It will contain too much code – most of the code is not used by most use cases.
• The copying of messages per client will not scale either.
• The client application could go away – in favor or configuration. Where a user wants to allow for dynamic configuration, the UI could be built specifically for that export service.
• It must be more adaptive to the use case. We need Export services to be created from a template to include elements like above, but not have all the options in one service.
• Essentially, we need an SDK that allows the export service to be created with modules the developer fills in for the validation, filtering, transforming, etc.
  o The scaffolding and kick off these modules (which would default to no-op elements for some use cases).
  o We would need to define the API, input and output of each module (much as EAI does) so that they are easily connected into the scaffolding.
We could produce some default export services that do things like get the data to Azure, AWS, etc. (not unlike how we provide Modbus, BACnet, etc. device services today).

- The “SDK” could be done in multiple languages in the future.
- Any existing EAI tool could be used to provide the scaffolding and modules when it exists (example: Spring Integration when using Java).

- **Action:** Jim to provide copy in Wiki and via email reflector to the group. Homework for all is to be able to provide feedback next week and during F2F.

**Additional Open Items**

- Keith – present idea to board that EdgeX sponsors a large demo to be used at shows/events. Tender to members to put the demo together and support it.