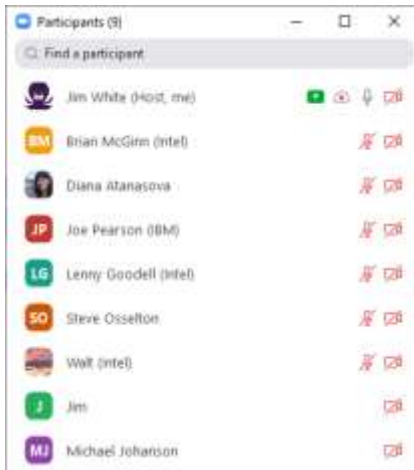




8/7/20

Attendees:



Others may have been in attendance after roll was captured.

Old Business

Current Status (Project board - as of 8/5)

- Backlog: 15
- Bugs: 4
- In progress: 3
- Under review: 0
- Done: 35
- New: 1

	Closed Issues	Working Issues (new)
edgex-go	2596, 2626, 2638, 2648	2609, 2643 (ENTRYPOINT) , 2647(validation logic profiles) , 2651 (sys mgmt CVE)
contracts mod	259, 249	258 (updating API) , 259 (remove base response)
bootstrap mod	85	

- NOTE: for brevity, “cak” = contracts; more precisely, an abbreviation for go-mod-core-contracts.
- 2597 issue: Invoke Validate function when adding yaml DeviceProfile (**done**)
 - **Removing this from old biz**
- API V2
 - Event handler / controller PR ([#2610](#))
 - Additional Lenny cleanup
 - Address comments of Redis implementation
 - **“Days away from completion”**



- Encounter some issues with Reading object - has inheritance which is not easy to do/use in Go. Cloud and team working
 - See go cak - #262
- Metadata DTO
 - PR [253](#) on core-caks (done and merged)
 - PR ([#2627](#)) - (done and merged)
 - How to differentiate the validation logic between Add and Update DTOs
 - New issue to deal with updating API ([#258](#) on core-caks)
 - Separate PR forthcoming that will present update DTO and how to process validate. Reaction from Core WG sought when it is out.
- Startup configuration ignored (duration & Interval)
 - PR [86](#) (bootstrap) merged (done)
 - Removing from old biz
- Do we really want to update the Geneva compose files for new releases of Device Service and/or App Services?
 - Jim provided Readme with link to the release notes and indication of new dot releases that can be dropped in as replacements
 - Removing from old biz
- CLI
 - Roadmap doc
 - <https://docs.google.com/document/d/1tHxomWv5C1bz7LHvrOcbSwRMSXhvBngWYhFUv5fk4JY/edit?ts=5f1efac4>
 - Release with Hanoi is goal; 2 open questions for Hanoi
 - What will be the release artifact (if any)?
 - Decision:
 - We version/tag it 1.0 -> indicating that it works with Hanoi
 - Don't release right now - let's make sure it is worthy of its 1.0 tag
 - Doesn't have to be released right at Hanoi, could be around Hanoi release (like DS, App Services, UI, etc.)
 - It will be included in the snap
 - Create executable artifacts and make them available via Github assets.
 - Rationale: in other CLIs, you don't have to build your own executable
 - Won't do it for every environment but for Windows, MacOS, Linux (exe vs linux artifact)
 - For both Intel and ARM (where applicable)
 - Side note: should we do something similar for C SDK?? Something to consider by TSC
 - DevOps to be consulted (Jim to address with Ernesto)
 - Needs to be part of DevOps - generate nightly
 - Github actions could be the implementation route



- Docker container could be an option in the future but not as part of “crawl” phase - would need to be on the network; then shell in
- What features are needed for this release?
 - Decision
 - API V2 - optional, but nice to have
 - Several services may have the same operation but the CLI does not let you specify the specific service (examples: device, ping, commands, etc.)
 - Option 1 - use a flag to specify a specific service
 - Option 2 - make the first positional param the service with second param the command (endpoint)
 - Example: metadata device list
 - Versus: device list
 - Clean up of --help documentation
 - Provide for complete core & support API Coverage
 - Jim to provide a gap analysis list
 - Core, metadata, command - must have; support nice to have
 - Note: core contracts clients doesn't cover all operations; use REST API instead - and in some cases we are doing this already
 - Consistency of operations with regard to JSON, YAML, TOML
 - Jim to provide gap analysis list
 - Be able to configure what services are addressed/included in status calls
 - Use some sort of configurable list

New business and UI direction were not addressed in the meeting due to lack of time. Will be addressed in 2 weeks.

NO meeting next week (Aug 13th)

New Business

- What should the Path for V2 Version endpoint be? (Lenny)
 - In V1 it is **/api/version**
 - For V2 it is specified as **/ap/v2/version**
 - V2 Response is same as V1 with addition **apiVersion**
- Issue [2651](#) - sys mgmt agent - use of Python 2 and vulnerability report

UI Direction

- [Roadmap deck \(draft\)](#)
- Demo
- Purpose (future - proposed): a production level user interface to manage and monitor a single instance of EdgeX Foundry (secured or unsecured)



(plus system management and UI)

- To manage and monitor from on or off box
- To monitor the state of EdgeX (status of services, memory, CPU usage, etc.) and provide alerts when something is outside of normal operating parameters
- Coming (Hanoi)
 - Manage app services (add functions, see data temporarily, etc)
 - Manage Kuiper rules
 - Documentation
 - “Make sure it is usable with what we have in EdgeX today”
- Roadmap items (past Hanoi)
 - See deck

Kubernetes Direction (release target is just a suggested goal)

- Crawl (Hanoi or Ireland): Have an example deployment.yml and service.yaml for deploying a single instance of EdgeX (minus device services) to a pod. With Redis, minus security.
 - Include example of how to setup device-virtual to send data to K8s instance
 - Document/demonstrate/provide example of setting up in K3s
 - Document what would not work in this environment
- Walk (Ireland): Include security
- Jog (Ireland or Jakarta): Helm charts & / or Operators
- Run (Jakarta or Kamakura): start addressing HA concerns in EdgeX