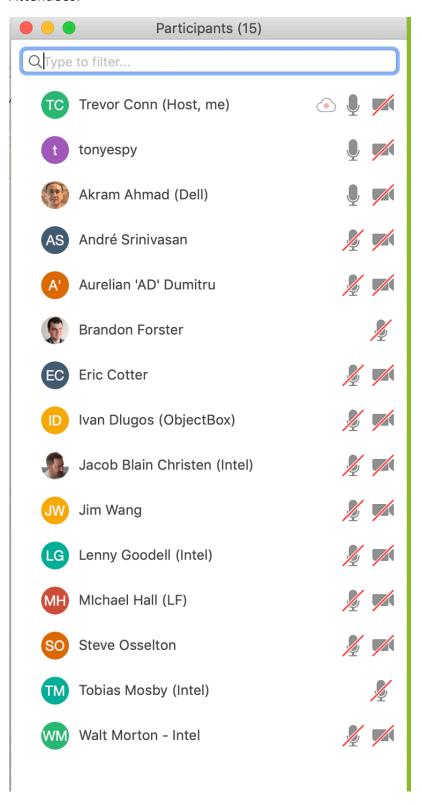# Core Working Group Meeting Notes (14-Feb-2019)

Attendees:

Old Business

- Plugins
  - Any possible updates from Beau's (Intel) contacts at Microsoft?
    - *Still waiting AFAIK*
    - *Will ping Beau again*
- Redis
  - Check in w/Andre RE: timeline – on track for 15-Mar, etc?
  - *We're still on track*
  - *Also working with IOTech on impl for notifications service*
- Modules
  - Initial population of go-mod-core-contracts complete
  - Lenny also has a draft of go-mod-registry, expect that PR shortly
    - *Tuesday next week*
  - Timing to discuss
    - Creation of go-mod-messaging (depends on core-contracts)
      - *On Dell's plate (#2)*
      - *Integrate with edgex-go (#3)*
    - Conversion of edgex-go (#1)
      - *Timing TBD (end of week possibly)*
    - Consumption of core-contracts by device-service-sdk
      - *If Steve is on*
      - *Trevor will create an issue in the device-sdk-go*
    - Release tagging
      - *Modules to be tagged in master (v0.1.0 for example)*
        - *Dependencies between modules*
        - *Messaging → core-contracts*
      - *Create Edinburgh branch*
        - *Initial PR against all deployed services to update go.mod pointing to module tags*
      - *Summarize versioning toward release timeline for review prior to Edinburgh (Trevor)*

New Business

- Float representations (via Tony Espy)

- o Seeking input from wider audience to build concensus
- o Options include
  - ▪ - change the default encoding of floats to use the C print floating point format { printf("%1.8e", 3.14159562) } which results in a string value like this: "3.14159265+e00"
    (note - for float64, the precision needs to be increased to 10+, which makes the logic a bit trickier)
  - ▪ - change the default encoding of floats to use some other literal representation of the actual binary value such a binary literal format as mentioned [here](#) (note, we'd have to implement this as the link points to a proposal for Go)
  - ▪ - keep the existing base64 encoding of floats and add an optional encoding attribute to the value property and value descriptor objects. This would allow alternate encodings to be added to handle floating point numbers
- o Preference is to keep base64
- o Possible impacts on application/export services
- o *Dependency on RulesEngine handling of float readings as base 64*
- o *TSC vote to finalize driving encoding via configuration*
  - ▪ *Base64 vs string w/precision*
  - ▪ *Proposed Edinburgh deliverable*
- Correlation ID demo if time permits
- *Question RE: Edinburgh scope – RulesEngine replacement? Not going to happen.*