Core Working Group Agenda (08-Nov-2018)

Attendees:

<table>
<thead>
<tr>
<th>TC</th>
<th>Trevor Conn (Dell) (Host, me)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>Rodney Hess</td>
</tr>
<tr>
<td>AH</td>
<td>Akram Ahmad (Dell)</td>
</tr>
<tr>
<td>A</td>
<td>Alberto</td>
</tr>
<tr>
<td>AF</td>
<td>Andrew Foster</td>
</tr>
<tr>
<td>BF</td>
<td>Brandon Forster</td>
</tr>
<tr>
<td>BH</td>
<td>Bruce Huang</td>
</tr>
<tr>
<td>D-</td>
<td>Drasko - Mainflux</td>
</tr>
<tr>
<td>EW</td>
<td>EdgeX Working Group 2</td>
</tr>
<tr>
<td>EA</td>
<td>Emad Attia (Intel)</td>
</tr>
<tr>
<td>EC</td>
<td>Eric Cotter</td>
</tr>
<tr>
<td>FT</td>
<td>Felix Ting</td>
</tr>
<tr>
<td>i</td>
<td>ijohnson</td>
</tr>
<tr>
<td>JW</td>
<td>Jim White (Dell)</td>
</tr>
<tr>
<td>SW</td>
<td>Sean Williams</td>
</tr>
<tr>
<td>TF</td>
<td>Tom Fleming</td>
</tr>
<tr>
<td>t</td>
<td>tonyespy</td>
</tr>
</tbody>
</table>
Old Business

- Follow-up action items from last week
  - Blackbox test PDF documentation has been added here
    - [https://github.com/edgexfoundry/blackbox-testing/tree/master/docs](https://github.com/edgexfoundry/blackbox-testing/tree/master/docs)
  - Walkthrough video recorded by Trevor for local Postman testing
    - [https://www.youtube.com/watch?v=LQU0Xf6EN0U](https://www.youtube.com/watch?v=LQU0Xf6EN0U)
- Discuss additional logistics (action items) around blackbox test move
  - If no owners for each step, then poor man’s method
  - Here are the steps
    - Move tests from blackbox repo to edgex-go (*Trevor*)
      - Define structure within edgex-go repo
        - Test definitions
        - Test execution scripts
        - Documentation
    - CI/CD changes (*?? With Jeremy leaving*)
      - Repointing of jobs
    - Need to establish a timeline for the movement and track it
      - Should we make this a regular item in the QA/DevOps WG call?
      - *Request that we keep agenda item on the above call as a check-in*
    - Either way, Contributor Guidelines need to be updated
      - Any supporting documentation (docs, videos) will need to be updated
      - *Probably need to update Contrib Guidelines w/poor man’s method for now*

New Business

- Finalize decision on ScheduleEvent property modifications
  - Separate “Owner” property
  - “Service” renamed to “Target”
  - *We are good to go to start making changes here.*
    - Past Core WG meeting notes include documentation of the decision
    - Tony to create issue for Device Service repos
- Snaps / SMA
  - *Delhi issues*
    - Doesn’t support working with Snaps via stop/start operations
    - Lack of error checking (false OK returned to client)
    - [https://github.com/edgexfoundry/edgex-go/pull/772](https://github.com/edgexfoundry/edgex-go/pull/772)
      - Comprehensive fix
    - [https://github.com/edgexfoundry/edgex-go/pull/750](https://github.com/edgexfoundry/edgex-go/pull/750)
      - Smaller footprint fix
- **Question is to which/whether to include in Delhi dot release**
  - Jim/Akram to discuss offline
  - Additional discussions in SMA meeting next week
- **Database Abstraction, remove Mongo types**
  - **DEMO: Brandon Foster**
    - *First step = change bson.ObjectId type in all models to string*
      - Push any ID generation down to Mongo provider layer
    - **Two questions**
      - Are we OK exposing internal IDs to outside world? Is there any downside to this?
      - Is there a use case for external apps to give us an identifier (UUID)?
    - **Mongo capability research**
      - Can BSONIDs be converted to GUIDs. Is there an equivalency? (TREVOR)

  \[ID = \text{Key (string)}\]

  \[Redis = \text{string (UUID)}\]

  \[Mongo = \text{BSONID}\]

  \[MySQL = \text{long}\]

- **Points from Tony**
  - BurntSushi lib doesn’t have official release, we’re using SHA right now. Concerning.
  - Same for Mongo, it’s out of date. Do we have a replacement?
    - Not right now. This is driving the database abstraction work.

- Discuss exposure of db generated keys outside of service boundary
- Keys versus UUID in usage
  - Key, represented as string – value is DB specific, cast by provider
    - Mongo compatibility
  - UUID is a standardized unique identifier
    - Could potentially be supplied by upstream application, not database.
- Go libraries for UUID generation
  - [https://github.com/google/uuid](https://github.com/google/uuid)
** my personal preference **
- https://github.com/satori/go.uuid
- https://github.com/nu7hatch/gouuid (OLD)

Supplemental notes

- DB → Business Layer ID representation
  - Answer question → UUID equivalency w/BSON ID
  - If not equivalent
    - Key (string type) DB PK value
    - UUID → std library type
      - Modify query paths accordingly
      - Indexing in Mongo for lookups

UniquelIdentifier (GUID) **

Key (BSON) – DB Key ** may not want this exposed **

Both will be strings as raw data types