Data Persistence Project Group (Inaugural) Meeting – 8/14/18

Attendees: TBD. Attendees that may have joined after the start of the meeting may not have been captured and listed.

Discussion and action items as a result of meeting in RED

Old Business

- New meeting time was based on consensus vote – any objections (apologies to those on West coast)
- Continuation of requirements discussion
  - NOV requirements (oil&gas)
    - 5 to 50K writes per second (depends on use case / device)
    - Majority are between 100-3000 on average writes per second
    - Reads are 10-20% of writes; while not a large number of reads, these do need to be efficient
    - Size of database on disk is not really an issue
    - RAM and CPU usage are the biggest concerns (Mongo gave them problems on these)
    - Startup time was also a concern (again Mongo problem)
    - Data is stored for up to 6 months on the edge (they have means to deal with too much or lost data at the edge)
    - Prefer to not embed the database – prefer flexibility to change and to distribute elsewhere
    - They don't have binary data, but use protobuf for Readings.
  - Hitachi Ventara
    - high ingress data @0.1 sec/message
  - Per last meeting
    - RFID Use case: 100+ / sec sensor reads
    - Building automation use cases: 1000+/sec
  - <u>Current Must haves</u>
    - License of product – compliant with Apache 2
    - Store and forward needs
    - Platform support:
      - Intel , ARM 64 bit
      - OS support: Linux, Unix, Windows, MacOS – those EdgeX has targeted
    - Performance
      - Need a holistic view. Small but slow is not acceptable.
      - Memory size, footprint, CPU, network? What is size? All these must be taken into consideration together.
      - Typically Prioritize writes over reads in performance
      - Chandra will provide some data to Jim
      - Concern impact of backup processes
    - Durable across EdgeX shutdown
    - Run in a container (Docker/Kubernetes/Snap/etc.)
      - Has to manageable from one control plane

      - Embedded might be considered by some, flexibility is more important to most (ability to easily swap out the database with proper app abstraction), and ability to distribute to alternate platform from micro services
    - Secure
      - Password protected
      - Supports data encryption (protect data at rest)
      - ???
    - Has Java, Go, C, C++ drivers/connectors
      - What others are probably needed
    - Community support and user-base size
      - How to quantify?
    - Binary support (as long as we can identify what type of binary)
      - Max size (up 16MB)
    - ???
  - Nice to haves
    - NoSQL (versus SQL)?
    - Synch capability?
    - Backup support?
    - Transactional (ACID) or Eventual Consistent (which CAP axis)?
    - Support multi-tenancy?
    - 32 bit support (Intel or ARM)
    - Database that also runs in memory
    - ??
- Should we start collecting potential options and can someone lead that effort
  - Not a review or evaluation at this stage, just a list of potential databases
    - MongoDB
    - Mongo Mobile
    - Redis
    - Influx
    - Cassandra
    - Couchbase
    - SQLite
    - Etc.

New Business