

Applications Services Face-to-Face Meeting

Boston, MA – Jan 24-25th

Dell EMC, Cambridge

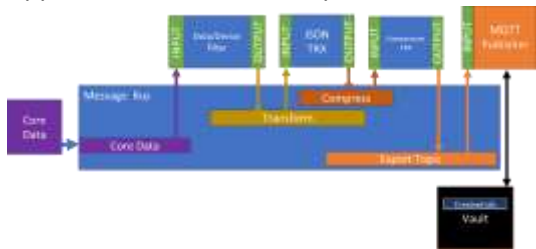
Proposed Agenda

Thursday, 9am

- Introductions/approval or adaptation of the agenda

9:30am – review of the current state, rationale for change, current high-level thinking

- Export Services Review
- Non-scalable solution and other issues
 - Can't be all things to all people
 - Too big and not easily customized
 - Copying event per client won't scale
- Application Services concept



10:30am – Ideation and Architecture

- Architectural Discussions
 - Message Bus
 - Which one?
 - Requirements, tradeoffs, selection discussion
 - Message Broker or Broker less?
 - What's the allowance impact to EdgeX footprint?
 - Brokered options include: Mosquitto, Paho, Apache ActiveMQ, Apache Apollo, RabbitMQ, Qpid, NATS
 - No Message Broker options include: OMQ, Nanomsg, GRPC, Thrift
 - Functions collected in a single "micro service" or just hang off the bus?



- How do we support multiple application needs and multiple endpoints (I want to send JSON to Azure but XML to my enterprise server)?
- FAAS (if not now, how do we prepare for possible FAAS in the future)

- What's the list of functions (long term) needed?
 - Filter (only give me readings from device A; only give me readings regarding temperature, ...)
 - Validator (device ID, reading against value descriptor, ...)
 - Transformation (convert C to F values, convert CBOR to Protobuf, ...)
 - Enrich (add device metadata to reading, ...)
 - Format (JSON, XML, CSV, ...)
 - Encrypt (really different kind of transformation)
 - Compress (really different kind of transformation)
 - Custom (black box that you define what you want to happen inside)
- Security
 - What needs to be secured? (Endpoints, encryption function?)
 - How are secrets provided and managed?
- Impact to platform independence?
- Can the function micro services be language independent?

3pm – Implementation Specifics

- Design Details
 - Message infrastructure configuration
 - Preconfigured channels/topics
 - How accomplished?
 - Any abstractions to the message infrastructure needed?
 - APIs for the “functions”
 - Interface definition sufficient to let developers start work
 - Data hand off between functions – what is the schema and format of the object going through the pipeline?
 - How and what do we secure?
 - Endpoint connection security (tokens to cloud products)
 - Data encryption secrets/certs
 - Core data – any changes necessary?
 - ZeroMQ replacement?
 - Implementation tools/framework considerations
 - Go, GoKit, ...
 - How do we support EdgeX internal local analytics (rules engine)?
 - How do we address CBOR payloads?
 - Is this a future consideration?
 - If so, do we need to refactor export services?
 - How do we handle monitoring and instrumentation?
 - What is needed to integration with System Management?
 - How do we test these new functions and services as a whole?
 - Integration tests?
 - Black box testing
 - How do we containerize the infrastructure and micro services?
 - Impact to CI/CD

- What is MVP for Edinburgh
 - Add additional subscriber to ZeroMQ
 - Potential message bus containerization
 - Specific MVP micro service functions to be implemented for Edinburgh
 - Suggested: filter, transform to JSON, compress
 - Specific MVP endpoint distributors to be implemented for Edinburgh
 - Suggested: MQTT endpoint
 - CI/CD jobs
 - Blackbox tests
- Who is doing what work for Edinburgh
 - Task assignments
 - Immediate action items
 - Issue creation in Github

Friday, 9am

- Pick up on any unfinished business from the previous day
- Future Considerations
 - Which cloud providers must we support going forward?
 - Azure IoT
 - AWS IoT
 - Google IoT Core
 - ???
 - What types of endpoints will we support going forward?
 - HTTP(s)
 - MQTT(s)
 - AMQP
 - XMPP
 - WebSockets
 - CoAP
 - Specific Cloud providers (above)
 - What functions are next after MVP functions?
 - Is there a call for help that we can make to the community to accelerate some of these?
 - Can we call on various community application and standard providers to help if we have them well documented?
 - Haystack
 - SAP Hana
 - ...
 - Can we architect this for SDK like behavior in the coming releases? What would an SDK look like? What would it do?
 - How can we make it simple(r) for users to construct new application services from a collection of existing functions?
 - How can we make it simple(r) for users to construct new functions and include them in the pipeline?

- Would there ever be a need for a client like we have with export services? Could an application service be dynamically programmed or configurable?
- What parts of this architecture are subject to change or user customization? Where do we need to provide more loose coupling going forward?
- How do we support inclusion of commands going forward?
 - Ex: How can we provide Azure IoT or some endpoint with command API details for actuation?
 - EdgeX Roadmap item: how to provide actuation context to the north side.
- Can we think about QoS levels in the future? How would that be incorporated?
- Tasks, action items review