Edinburgh Release

Code Freeze: May 28, 2019

Release: June 20, 2019 (ver 1.0.1 - July 22, 2019)

- Release Themes and Objectives
- Application Working Group Tasks and Notes
- Core Working Group Tasks and Notes
- Device Service SDK and Device Services Group Tasks and Notes
- Security Group Tasks and Notes
- System Management Group Tasks and Notes
- Test/QA/Documentation Group Tasks and Notes
- DevOps Group Tasks and Notes
- General Tasks and Notes

The Edinburgh release is expected to be version 1.0 and a significant milestone in the EdgeX progression. It will be made available shortly after the 2nd anniversary of the project and will signal EdgeX is fit for purpose and ready for wide use/dissemination and long term support in edge/IoT solutions. The Edinburgh release will focus on an improved user on-boarding experience, support for binary data, better database/persistence abstraction (for future database replacements), testing dashboard and performance testing, the initial implementation of an eventual export service replacement (application services) in addition to additional security and system management needs that were started in 2018.

Release Themes and Objectives

- Improved on-boarding for EdgeX users. This includes documentation, tutorials, dev kits, and other material to make getting, using and understanding EdgeX easier.
- EdgeX will support the ingestion, use and export of binary data in CBOR format. Device Services will be allowed to send binary data as part of the Event/Reading packages. Core and Export Services will be adapted to handle, persist, and route/send binary data as it does integer, float, and string data today. Incorporation of binary data will allow custom built local analytics to use the binary data to trigger actuation of devices. https://github.com/edgexfoundry/edgex-go/issues/820
- EdgeX database-using services (Core Data, Metadata, Export Client, Logging, Notifications, and Scheduling) will be refactored to be more loosely coupled to the persistence mechanism (currently MongoDB). This will ease the use of alternative persistent stores and technology (such as streaming) in future implementations and even allow the project to select alternate or additional reference implementation databases in future releases. https://github.com/edgexfoundry/edgex-go/issues/821
- The current Export Services, while functional, create scalability issues. The current EdgeX capability relies on the Export Services to know and understand all possible endpoint distribution mechanisms and transformation capability necessary to support all clients – even if the use case requires only a single, simple client need. As the number and type of EdgeX north side clients expands, this service will be too big and complex to support the north side needs. The initial and simple Application Services in the Edinburgh release will be designed and implemented to support smaller, more tailored exportation needs. These services will contain just the functionality (filtering, transformation, enrichment, etc.) needed for a single endpoint. To address multiple endpoints, multiple Application Services will be created. In a way, the Application Services will begin to look more like the south side Device Services – that is functionality dedicated to a particular client protocol and data need. Long term over the course of a number of releases, Application Services will replace EdgeX Export Services as the north side distribution facility. https://github.com/edgexfoundry/edgex-go/issues/822
- Provide many Device Service implementations using the Delhi release provided Go and C DS SDKs. As a target, the Edinburgh release will provide replacement services for the existing Java Modbus, BACNet, BLE, MQTT and SNMP device services. Many additional Device Services are anticipated with the Edinburg release. https://github.com/edgexfoundry/edgex-go/issues/823

Application Working Group Tasks and Notes

- The Edinburgh release will feature a new service type called Application Services as indicated in the major themes and objectives of this release that will be the eventual replacement for Export Services. Application Services may include experimentation in technology such as GoKit and Function-as-a-service (aka serverless compute) to help implement. https://github.com/edgexfoundry/edgex-go/issues/822

Core Working Group Tasks and Notes

- Per the release themes and objectives above, the services using the database (currently MongoDB) will be refactored to have a better database abstraction architecture and allow for different database persistence in the future. This includes removing BSON references, hiding domain IDs, providing an abstraction to allow for object identification that is transformed across platforms. Changes will be made to appropriate core, support, and export layer services. https://github.com/edgexfoundry/edgex-go/issues/821
- Given Go Glide (used for versioning and dependency management) is being deprecated, the project must begin to move to an alternative. The Go community is adopting Go modules (formerly vgo) as its replacement. In this next release, the project will move to Go modules for Go code dependency management. https://github.com/edgexfoundry/edgex-go/issues/477
- Addition of a correlation id that is added to the event/reading created by the device services and used throughout the rest of EdgeX starting with the core services. A correlation id allows for tracing of a sensor reading across services. Tracing is the ability to trace a system API request inside and across EdgeX services for the purpose of debugging and performance metrics tracking. Tracing is provided by frameworks such as GoKit, but EdgeX has not yet adopted such a framework and needs a solution that is language independent (for use across non-Go services like C based Device Services). https://github.com/edgexfoundry/edgex-go/issues/824
- Scheduler service (and the metadata data) will be altered to associate schedule events/schedules to a particular service owner and allow the Scheduler Service to exercise those “owned” by the Scheduler Service on the appropriate service. https://github.com/edgexfoundry/edgex-go/issues/796
- Edinburgh release will provide the means to start Consul so that the configuration information is preserved and config-seed does not repopulate (or overwrite) the configuration already in Consul. This will allow the setting of configuration by the SMA. https://github.com/edgexfoundry/edgex-go/issues/827

Device Service SDK and Device Services Group Tasks and Notes

- The additional of several Device Services as stated above. The list includes Modbus, BACnet, BLE, MQTT, SNMP. The Device Services may be platform specific (Linux vs. Windows for example) and use various platform specific features (such as the BlueZ Linux Bluetooth stack). Alternate implementations or implementations that are more generic to the protocol will always be encouraged to be provided from the community and third parties. Additional Device Services may be provided (and encouraged by the project) at the contributor's discretion. As an additional note, all Java Device Services will be archived with the Edinburgh release as they no longer function with the changes. https://github.com/edgexfoundry/edgex-go/issues/823
- The issue of device ownership or management of devices will be corrected in the Edinburgh release. Metadata will be considered the source of device knowledge for all of EdgeX and with the Edinburgh release, any deleting a device in metadata requires removing all associated metadata information in metadata as well (provision watcher, etc.). Additionally, a blacklist of devices (managed in and by Core Metadata) will be added and any time a device is deleted, the device will be added to the blacklist so that the device does not automatically get provisioned again when auto provisioning of the device is triggered by a device service. https://github.com/edgexfoundry/edgex-go/issues/828
- To better support their use and to help drive the production of more south-side connectivity, this release will include Device Service SDK tutorials, examples, and how-to guides.
- The SDKs (and resulting Device Services) will offer the system management APIs and be fully integrated into the EdgeX system management capabilities. https://github.com/edgexfoundry/edgex-go/issues/794
- Improve/simplify the Device Profile – removing unused elements and better naming of elements given recent SDK work. https://github.com/edgexfoundry/edgex-go/issues/830
- The Addressable, used in multiple areas of EdgeX to identify a service or data destination such as a REST endpoint or MQTT topic, will be relooked and potentially modified to better support the broader needs of project. https://github.com/edgexfoundry/edgex-go/issues/831

Security Group Tasks and Notes

- Edinburgh will include automated testing of security features (currently tested manually). https://github.com/edgexfoundry/security-api-gateway/issues/44 & https://github.com/edgexfoundry/security-secret-store/issues/51
- With Vault in place to store EdgeX application secrets, the Edinburgh release will include and use Vault namespaces for storing secrets for various services and the release will have some service(s) get/store their secrets in Vault to secure them. Future releases will require all application secrets (currently stored in configuration) to be stored/obtained via the secure store. https://github.com/edgexfoundry/security-secret-store/issues/52
- By the Edinburgh release, service to service AuthN/AuthZ requirements will be documented and a preliminary design will be presented for community reaction.
- EdgeX will explore moving to Kong and Vault latest (0.13.0 for Kong) and possibly move to the latest if it is not disruptive in order to keep up with the latest versions of the 3rd party tools as a general rule. https://github.com/edgexfoundry/security-secret-store/issues/53 & https://github.com/edgexfoundry/security-api-gateway/issues/45

System Management Group Tasks and Notes

- In this release, the System Management Agent and associated system management API will add the ability to track service CPU usage and metrics. https://github.com/edgexfoundry/edgex-go/issues/832
- The SMA will add an API to provide the health/status check of a service, but this will be a proxy to the configuration/registry service. https://github.com/edgexfoundry/edgex-go/issues/833
- The SMA start/stop/restart capabilities will be refactored to better address Docker environments as well as generic executable calls. https://github.com/edgexfoundry/edgex-go/issues/834

Test/QA/Documentation Group Tasks and Notes

- With automated blackbox tests in place, the Edinburgh release will include better visualization/dashboard of test results (including look up and display of historical test results). Candidate tools include Telegraf + Grafana + InfluxDB, DataDog, Prometheus. https://github.com/edgexfoundry/blackbox-testing/issues/105
- The Edinburgh release will include the capture of resource metrics to monitor performance. Metrics monitored will include memory and CPU consumption. https://github.com/edgexfoundry/edgex-go/issues/115
- As a stretch goal, the Edinburgh release will include performance/load testing using a tool like Bender, Jmeter, Load Impact or other tool selected by the work group.
- Security tests will be automated as mentioned above in the Security Group tasks. https://github.com/edgexfoundry/security-api-gateway/issues/44 & https://github.com/edgexfoundry/security-secret-store/issues/51
- For the Edinburgh release, code contributors will now be required to supply additional or updated blackbox tests to cover changes made to the code base with any PR. Working group leads and project committers are head accountable for this change in procedures.

DevOps Group Tasks and Notes

- DevOps Chairman is resigning. New leadership is needed to guide this work group. Over the course of the next couple of months, the current chair will hold review sessions during the working group meetings to educate more of the community or current systems and procedures and find a volunteer to guide and steer the direction of this important working group going forward. The target is to identify a new leader by Christmas time 2018.
- EdgeX Go code (inclusive of all development, build and test environments) will move to Go 1.11. https://github.com/edgexfoundry/edgex-go/issues/764 & https://github.com/edgexfoundry/edgex-go/issues/765
- EdgeX Go code will use modules in place of Glide. https://github.com/edgexfoundry/edgex-go/issues/477
- EdgeX will update to Consul 1.2.3 for Edinburgh. https://github.com/edgexfoundry/edgex-go/issues/836

General Tasks and Notes

- Given Edinburgh will be EdgeX version 1.0, the EdgeX community will adopt a release, support and version strategy to provide guidance around backward compatibility to the development and user community. Additionally, the community will define a long term support policy that defines the communities intended support of the EdgeX Foundry platform.
- EdgeX has used Rocket Chat as the message channel for contributor and user information exchange since the project's inception. With the upcoming release, the community is going to switch to Slack – a more widely used and accepted form of developer communications.
- With the Edinburgh release, the community has resolved to adopt a "Release Manager" to monitor, guide, document and manage each release. The Release Manager role will rotate through organizations like Dell, IoTech, Intel, and Canonical. Michael Hall will lead efforts to define the duties of the role and Keith Steele (or TSC chair) will coordinate the rotation of the role to the larger supporting organizations.
- EdgeX will have and advertise its support contract/policy to the world with the Edinburgh release.
- The TestQA and DevOps working groups will meet simultaneously in order to stream line efforts that are often intertwined.

Release Summary Deck