

Delivery: ~ April 2019

The Edinburgh release is expected to be a significant milestone in the EdgeX progression. It will be made available right around 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.

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. [Keith , TSC chairman, initiative supported by Michael Hall and the entire project]
- 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. [effort to be led by Trevor Conn and the Core Services Working Group]
- 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. [effort to be led by Trevor Conn and the Core Services Working Group]
- 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. [effort to be led by Janko Isidorovic and the Applications Working Group]
- EdgeX has greatly reduced its memory and file size footprint, and improved the performance of many operations (to include startup time) over the past two releases. An automated performance framework will be implemented for the Edinburgh release to continually check that the performance metrics of EdgeX meet or exceed the expectations of an edge software platform as determined by the community. The current platform performance targets can be found here: <https://wiki.edgexfoundry.org/display/FA/California+Release#CaliforniaRelease-TargetPerformance>. [effort to be led by Andy Foster and the Test/QA Working Group]
- 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 Edinburgh release. [effort to be led by Steve Osselton and the Device Services Working Group]

- A certification program will be outlined in concert with the Edinburgh release. A certification program will allow third parties creating EdgeX services to verify their services as alternative or enhancing capability to those provided by the EdgeX open source effort. [effort to be led by Jason Shepherd's team]

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. [effort to be led by Janko Isidorovic and the Applications Working Group]
- Implementation of a lightweight rules engine option or other "edge analytic" capability to replace the Java, Drools-based rules engine currently serving as the EdgeX reference implementation. This is the last of the Java micro services still used in EdgeX. [effort to be led by Janko Isidorovic and the Applications Working Group]

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. [effort to be led by Trevor Conn and the Core Working Group but also incorporate other working groups for their services after prototyping]
- 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. [effort to be led by Trevor Conn and the Core Working Group to start and prototype]
- Tracing is a stretch goal task for the Edinburgh release. 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). [effort to be led by Trevor Conn and the Core Working group if it is tackled in this release]
- 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. [effort to be led by Eric Cotter under Trevor Conn and the Core Working group management]
- Metadata will include a new device blacklist (with associated management API). This blacklist will allow devices to be excluded from automatic provisioning. See related item in the Device

Service SDK and DS Group Tasks. [effort to be led by Steve Osselton and the Device Services Working Group with consultation and help of the Core Working Group].

- 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. [effort to be led by Trevor Conn and the Core Working group]
- ~~Edinburgh release will improve resiliency and availability of the services by improving client code to automatically pick up/detect endpoint changes. [effort to be led by Trevor Conn and the Core Working group]~~

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. [effort to be led by Steve Osselton and the Device Services Working Group].
- 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. [effort to be led by Steve Osselton and the Device Services Working Group with consultation and help of the Core Working Group].
- 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. [effort to be led by Steve Osselton and the Device Services Working Group with help from Michael Hall and emphasis by TSC chair Keith Steele].
- The SDKs will implement a means to provide a cache of readings. This allows the collection and response for a request of a reading to be decoupled (and more asynchronous). [effort to be led by Steve Osselton and the Device Services Working Group].
- The SDKs (and resulting Device Services) will offer the system management APIs and be fully integrated into the EdgeX system management capabilities. [effort to be led by Steve Osselton and the Device Services Working Group with consultation and help of the System Management Working Group].
- Improve/simplify the Device Profile – removing unused elements and better naming of elements given recent SDK work. [effort to be led by Jim White to convene current and past SDK developers to document the current profile and Steve Osselton and the Device Services Working

Group with consultation and help of the Core Working Group that owns Metadata to define and implement desired changes].

- 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. [effort to be led by Steve Osselton and the Device Services Working Group with consultation of the Core and Application Working Groups]

Security Group Tasks and Notes

- Edinburgh will include automated testing of security features (currently tested manually).
- 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. [effort to be led by David Ferreira and the Security Working Group]
- The initial secrets (needed to start Vault/Kong) will be encrypted on the file system using a secure storage abstraction layer – allowing other implementations to store these in hardware stores (based on hardware root of trust systems). [effort to be led by David Ferreira and the Security Working Group]
- By the Edinburgh release, service to service AuthN/AuthZ requirements will be documented and a preliminary design will be presented for community reaction. [effort to be led by David Ferreira and the Security Working Group]
- 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. [effort to be led by David Ferreira and the Security Working Group]

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. [effort to be led by Jim White and the System Management Working Group]
- 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. [effort to be led by Jim White and the System Management Working Group]
- The SMA will be refactored to reduce or remove the inclusion of Docker libraries by build ID from system management services. [effort to be led by Jim White and the System Management Working Group]
- As a stretch goal, the SMA will provide a translation layer (implemented via necessary abstraction) to offer the SMA API (and associated data) via other protocols starting with one protocol (like LWM2M or SNMP). In effect, SMA will provide access to SMA API and control plane data in a fashion similar to how Export Services makes data plane available to 3rd parties in a fashion dictated by those 3rd party clients. [effort to be led by Jim White and the System Management Working Group]

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. [effort to be led by Andy Foster and the Test/QA Working Group]
- The Edinburgh release will include the capture of resource metrics to monitor performance. Metrics monitored will include memory and CPU consumption. [effort to be led by Andy Foster and the Test/QA Working Group]
- 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. [effort to be led by Andy Foster and the Test/QA Working Group]
- Security tests will be automated as mentioned above in the Security Group tasks. [effort to be led by David Ferriera and the Security Working Group]
- 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.
- As a stretch goal, the project will look to produce Swagger documentation to better support testing and API documentation in future releases. RAML documentation will remain the official API standard for Edinburgh, but additional Swagger documentation will be provided and weighed for use in future releases. [effort to be led by Cloud Tsai and the Test/QA Working Group]

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 on 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. [effort to be led by the chair of DevOps and the DevOps Working Group]
- EdgeX Go code will use modules in place of Glide. [effort to be led by the chair of Trevor Conn and the Core Working Group]
- EdgeX will update to Consul 1.2.3 for Edinburgh. [effort to be led by the chair of Trevor Conn and the Core Working Group]
- Program artifacts (Go service executables, Java JAR files, etc.) will be retained and made available in Nexus. This will allow 3rd parties to more easily reuse and package EdgeX for alternate deployments and to better support development efforts (not requiring all the services to be built all the time to work on a single service). [effort to be led by the chair of DevOps and the DevOps Working Group]

General Tasks and Notes

- 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. [effort

to be led by the chair of DevOps and the DevOps Working Group once approved and budgeted by the EdgeX board]

- 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. [effort initially led by Michael Hall to define the role and its duties for adoption by TSC]
- EdgeX will have and advertise its support contract/policy to the world with the Edinburgh release. [Jim White will draft a strawman support document and work the document through the DevOps working group]
- The TestQA and DevOps working groups will meet simultaneously in order to stream line efforts that are often intertwined.
- A stretch goal for the Edinburgh release is something like a “12 factor apps” for how developers can build EdgeX services to better survive and operate in a distributed environment. An additional stretch goal for Edinburgh would be to have a playground or test environment to test EdgeX service distribution.