Release scope decisions, architectural notes, design decisions

This page contains design and scope decisions relative to each release. Major design decisions (redesign something or design something new; architecturally significant decisions and/or cross cutting architecture concerns) are addressed in Architecture Decision Records (ADR).

This page is maintained by the EdgeX Foundry Product Manager and Release Czar with inputs from working group chairs.

Hanoi (~ fall 2020)

- Decision was made to remove client monitoring (in Core WG on 6/18 - see issue 2594)
- As it pertains to the V2 API, the decision was made that Gets (queries) for by ID be removed and have Gets by name instead. An example would be for get events by device id vs device name.
- With regard to IPv6, the EdgeX community has tested that EdgeX services will work with IPv6. However, configuration for Kong or device services running on a separate box have not been addressed for IPv6. The EdgeX community has not focused on overlay network (service to service on the Docker network). This is on the roadmap for Ireland or beyond.
- When an event (model instance usually done by a Device Service) is created, a UUID gets created and attached to the event no matter where it is created (per Core WG 8/20/2020). ADR forthcoming.
- In support of tagging and identifying the origin of event/reading data, a new field "tags" will be added to event to allow the system to add key/value pairs to the event to indicate its origin. It could be set with GPS coordinates, factory location, etc. For this Hanoi release, the tags field will be populated via the app service function just before export (see issue 270 on core contracts). In future releases, the tags may be populated by other implementations and by other services (such as the originating DS). This decision was made in the architects meeting of 7/20 and further discussed in the core WG of 8/20.

Ireland (~ spring 2021)

- Decision was made (per Monthly Architect's meeting of 9/21/20) to add configuration to allow default overrides for HTTP request maximum size and number of simultaneous requests.