

Edinburgh Release – 1.0

Strike Team

Binary/CBOR → EdgeX

Foundry

Jenko (TBD)

Mainflux

Trevor Conn

Senior Principal Software Engineer

Dell

Tobias Mosby

Senior Software Engineer

Intel Architecture, Graphics and Software (IAGS)

CBOR Strike Team

Recap Objectives

Goal: *EdgeX 1.0 will support the ingestion, use and export of binary data in CBOR format.*

Device Services to send binary data as part of Event/Reading.

Core and Export Services 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.

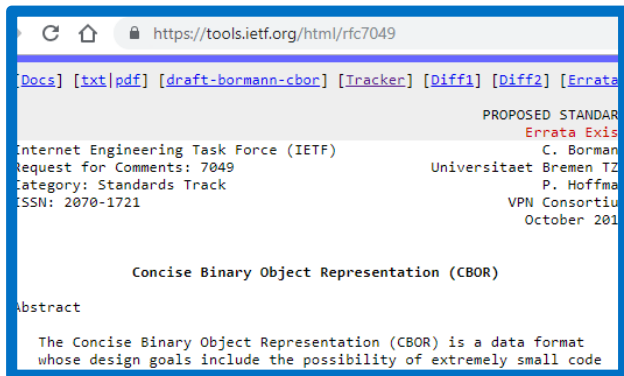
- Next Steps: Go and C/C++, Device SDK example events**
- Choose device service / application to exercise end to end flow
- Receiver to visualize the decoded binary events
- Readings export successfully to cloud



CBOR facilitates a standardized and efficient way to transfer binary/object data at scale

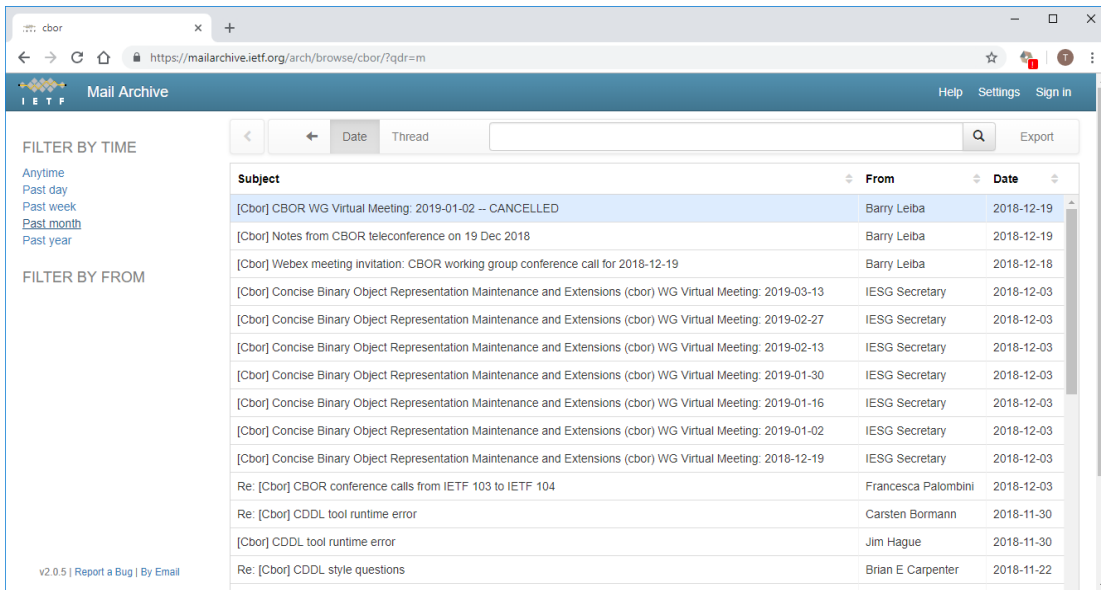
What is CBOR – IETF Standard

Spec: RFC7049



A screenshot of the IETF website showing the page for RFC 7049. The browser address bar shows `https://tools.ietf.org/html/rfc7049`. The page content includes navigation links like [Docs], [txt|pdf], [draft-bormann-cbor], [Tracker], [Diff1], [Diff2], and [Errata]. It identifies the document as a PROPOSED STANDARD and lists the authors: C. Bormann, Universitaet Bremen TZ, P. Hoffman, and VPN Consortium. The title is "Concise Binary Object Representation (CBOR)". An abstract is visible, stating: "The Concise Binary Object Representation (CBOR) is a data format whose design goals include the possibility of extremely small code".

IETF Mailing List



A screenshot of the IETF Mail Archive website for the CBOR mailing list. The browser address bar shows `https://mailarchive.ietf.org/arch/browse/cbor/?qdr=m`. The page has a navigation bar with "Mail Archive", "Help", "Settings", and "Sign In". There are filters for "FILTER BY TIME" (Anytime, Past day, Past week, Past month, Past year) and "FILTER BY FROM". A table of email messages is displayed with columns for Subject, From, and Date.

Subject	From	Date
[Cbor] CBOR WG Virtual Meeting: 2019-01-02 -- CANCELLED	Barry Leiba	2018-12-19
[Cbor] Notes from CBOR teleconference on 19 Dec 2018	Barry Leiba	2018-12-19
[Cbor] Webex meeting invitation: CBOR working group conference call for 2018-12-19	Barry Leiba	2018-12-18
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-03-13	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-02-27	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-02-13	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-01-30	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-01-16	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2019-01-02	IESG Secretary	2018-12-03
[Cbor] Concise Binary Object Representation Maintenance and Extensions (cbor) WG Virtual Meeting: 2018-12-19	IESG Secretary	2018-12-03
Re: [Cbor] CBOR conference calls from IETF 103 to IETF 104	Francesca Palombini	2018-12-03
Re: [Cbor] CDDL tool runtime error	Carsten Bormann	2018-11-30
[Cbor] CDDL tool runtime error	Jim Hague	2018-11-30
Re: [Cbor] CDDL style questions	Brian E Carpenter	2018-11-22

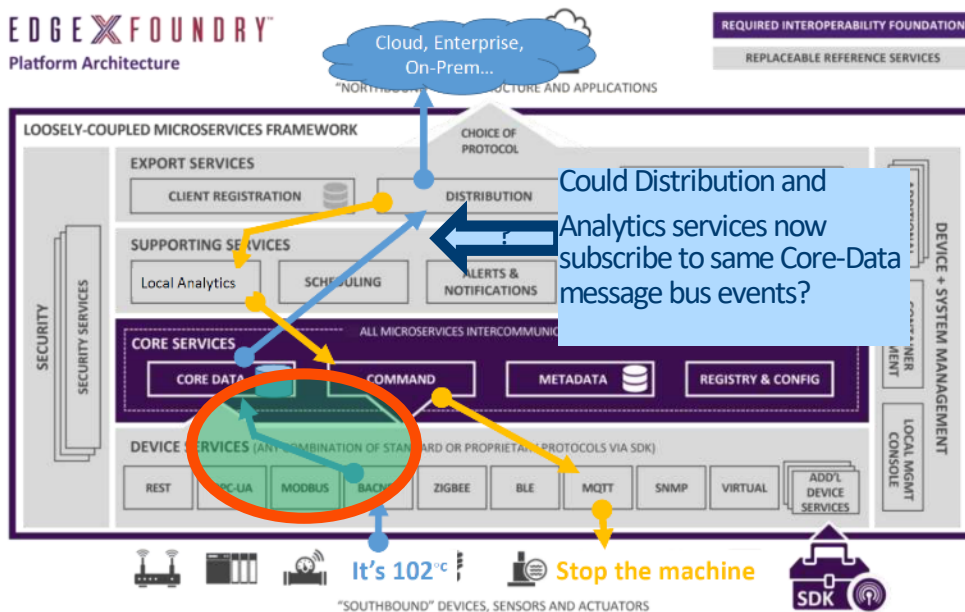
CBOR: Available Language Bindings

- JavaScript, NodeJS, Lua, C#, Java, Python, Perl, Rust, Erlang, Swift, ...
- Open Source – Permissive Licenses
- Libraries contain methods to aid conversion of JSON formatted strings or streams to standard library and/or user defined data structures (e.g., Go/C++ objects) and back.
- **Go:** [ugorii/go](#)
- **C/C++:** [intel/tinycbor](#)
 - TinyCBOR is Intel's industrial strength C/C++ implementation of CBOR, as used in the [IoTivity](#) framework
 - Qt implementation now includes a derived implementation

Integrating with EdgeX Event/Data Workflow

Initial POC scope focused on a simple **event**, sent from Device Service to Core Data

1. Transform data emitted by a virtual device, ~~temperature~~ light switch readings from device-sdk example
2. Convert Events/Readings information that is currently passed as JSON into CBOR equivalent.
3. Modify APIs within this workflow to properly send the information; e.g., including "application/cbor" MIME type. OMQ and/or REST endpoints (tbd)
4. Evaluate EdgeX design, such as responsibility of Export Services to handle transformation, formatting, filtering of Event data for subsequent transmission "north" (to enterprise/cloud systems)



POC EdgeX Events w/ Binary Value Descriptors

- Implements device-simple example with binary origin value. Added "NewBinaryValue" helper to populate CommandValue.binValue field as **CBOR-encoded** data.

- Command converts received {"Switch":"True"} to binary 00 or 01 emits event to EdgeX Core-Data

```
level=DEBUG ts=2019-03-01T17:49:32.005907093Z app=device-simple source=command.go:232 msg="Handler -
execWriteCmd: putting deviceResource: {\description:\Switch On/Off.\,name:\SwitchButton\,tag:\
"tag":null,\properties\:{\value\:{\type\:\Bool\,\readWrite\:\RW\,\units\:{\type\:\String\,\
readWrite\:\R\,\defaultValue\:\On/Off\}},\attributes\:\null}"
level=DEBUG ts=2019-03-01T17:49:32.007481682Z app=device-simple source=simpledriver.go:93 msg="Simple
Driver.HandleWriteCommands: device: Simple-Device01, operation: set, parameters: [Origin: 15514625726
00, Bool: true]"
level=DEBUG ts=2019-03-01T17:49:32.009019073Z app=device-simple source=command.go:107 msg="Handler -
execReadCmd: deviceResource: {\description:\Switch On/Off.\,name:\SwitchButton\,tag:\nu
ll,\properties\:{\value\:{\type\:\Bool\,\readWrite\:\RW\,\units\:{\type\:\String\,\
readWrite\:\R\,\defaultValue\:\On/Off\}},\attributes\:\null}"
level=DEBUG ts=2019-03-01T17:49:32.010522894Z app=device-simple source=simpledriver.go:50 msg="Simple
Driver.HandleReadCommands: device: Simple-Device01 operation: get attributes: map["
TJM: PUT OriginBytes:
01
level=DEBUG ts=2019-03-01T17:49:32.012432673Z app=device-simple source=executor.go:69 msg="Schedu
Event turnOnSwitch executed result- Event: <nil>, AppErr: <nil>"
level=DEBUG ts=2019-03-01T17:49:32.01363471Z app=device-simple source=simpledriver.go:55 msg="\nCur
nt SwitchButtonVal: true\n\n"
TJM: OriginBytes:
01
level=DEBUG ts=2019-03-01T17:49:32.017553333Z app=device-simple source=command.go:169 msg="Handler -
execReadCmd: device: Simple-Device01 RO: {\index\:\null,\operation\:\get\,\object\:\SwitchBut
ton\,\parameter\:\Switch\,\resource\:\null,\secondary\:\[],\mappings\:\{}} reading: {\origin
:\1551462572013,\device\:\Simple-Device01\,name\:\Switch\,value\:\n\n}"
level=DEBUG ts=2019-03-01T17:49:32.019226667Z app=device-simple source=executor.go:69 msg="Schedu
Event readSwitch executed result- Event: {\device\:\Simple-Device01\,\origin\:\1551462572019\,\r
eadings\:\{\origin\:\1551462572013,\device\:\Simple-Device01\,name\:\Switch\,\value\:\n\n
\n"}}, AppErr: <nil>"
```

- Past events are visible from this endpoint. Ready to be extended with **CBOR decode**.

Fetch Sample Events Simple-Device01
Get last 17 events emitted.

http://edgex-core-data:48080/api/v1/event/device/Simple-Device01/17 GET URL params Headers

Send Save Preview Add to collection Res

Body Headers (3) STATUS: 200 OK TIME: 127 ms

Pretty Raw Preview JSON XML

```
1 [
2   {
3     "id": "8afdd4dc-5296-461f-bb77-590ba8ac091d",
4     "device": "Simple-Device01",
5     "created": 1551462332024,
6     "modified": 1551462332024,
7     "origin": 1551462332019,
8     "readings": [
9       {
10        "id": "e102f5fb-413b-4e2d-b4f2-045301c8ed6d",
11        "created": 1551462332023,
12        "origin": 1551462332015,
13        "modified": 1551462332023,
14        "device": "Simple-Device01",
15        "name": "Switch",
16        "value": "\n"
17      }
18    ]
19  }
20 ]
21 {
22   "id": "9fb39e12-f6c8-421c-ba68-bc1blad21d88",
23   "device": "Simple-Device01",
24   "created": 1551462327016,
25   "modified": 1551462327016,
26   "origin": 1551462327013,
27   "readings": [
28     {
29       "id": "fd47d6e3-4883-4853-b448-2d86b3eb1612",
30       "created": 1551462327016,
31       "origin": 1551462327009,
32       "modified": 1551462327016,
33       "device": "Simple-Device01",
34       "name": "Switch",
35       "value": "\n"
36     }
37   ]
38 }
```

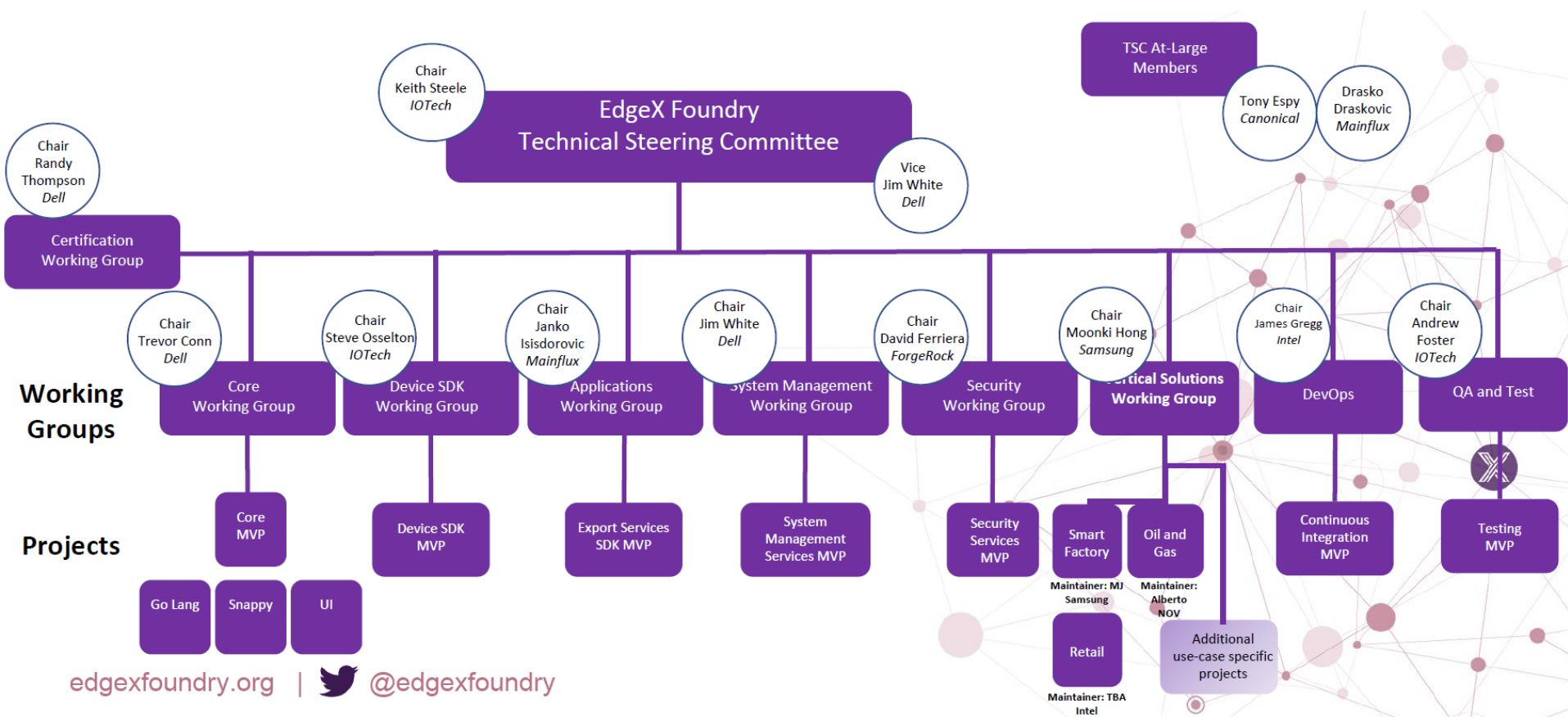
Populated by new CBOR-encoded ValueDescriptor.binValue field

Release Priorities / Readiness Challenges

- A lot of in-flight changes introduced several opportunities for design-time reconsideration:
 - Decoupling Mongo DB
 - Switch to Go Modules
 - Upstream error handling
 - Legacy Bugs
- Commerce WG is building solutions to relevant use cases.
- Methodologies to increase communication and tweak lifecycle.

Avoid Surprises - Communication is Key!

Some sources/targets of risk ...



Process Improvements / Constraints

- Utilize “Strike Teams” to:
 - Identify primary owners/contacts
 - Plan and resource remaining critical release features
 - Facilitate more regular sync / sessions.
 - Swarm to complete and communicate status
 - Designate adoption “handoff” owners – Strike Teams to file github issue tagged with criticality and contents that instruct when and how to adopt upstream changes.
- Adopt common way to capture EdgeX architectural decisions
 - Example: Lightweight ADRs
https://github.com/joelbarkerhenderson/architecture_decision_record

Strike Teams Driving Remaining Efforts

Strike Team	Owner
Security Features	Jim, Tingyu, (+ 1-2 Beau resourcing)
CBOR/Binary Throughout EdgeX	Toby (main design and DS implementation), Trevor (core/supporting changes), Janko's Mainflux resource (export & application services changes)
Redis Replacement Services	Trevor and Andre
Performance Testing	Andy and IoTech team
Unit Testing Coverage (esp SDKs)	Andy, Steve and IoTech team
Certification Program Definition	Randy and Jim
Documentation Cleanup	Open to Volunteers, (+ Keith and Jim seeking resources)

CBOR Strike Team Scope / Constraints

CBOR Strike Team members will identify and communicate details such as...

- Selection of a particular device service to serve as MVP strawman producer of CBOR encoded data.
 - **device-media-go** uses device-camera-go to identify IP cameras of interest (i.e., discovered and tagged to opt in). It populates an event Reading.binValue with CBOR encoded image data at designated interval.
 - **device-camera-go** adds getImage command directly to populate Reading.binValue with lowest resolution image.
 - **device-mqtt-go** transmits CBOR encoded data as alternative to JSON payload.
 - **coap-iotivity** ?

CBOR Implementation Scope/Constraints

- General

- Guidelines for global vs distributed configuration and scheduling? Fuji
- To define success criteria early. Should we implement tests first?
- Does community typically stub primary interfaces early – e.g., add new error type to support tests?

```
if hdrContentType.contains("application/cbor"):  
    return types.NewErrServiceClient(resp.StatusCode, []byte(ErrNotImplemented{}.Error()))
```

- Developer/Integrator opt-in to CBOR/binary transmission:

- Device service author explicitly adds binary reading values.
- Core-Data automatically identifies received binary data types in readings.
- Configuration driven; option indicates if *full device schema* event/readings payloads should be emitted as human readable JSON, Base64 encoded, CBOR encoded.

Add 'name' param to

Toby NewBinaryValue helper

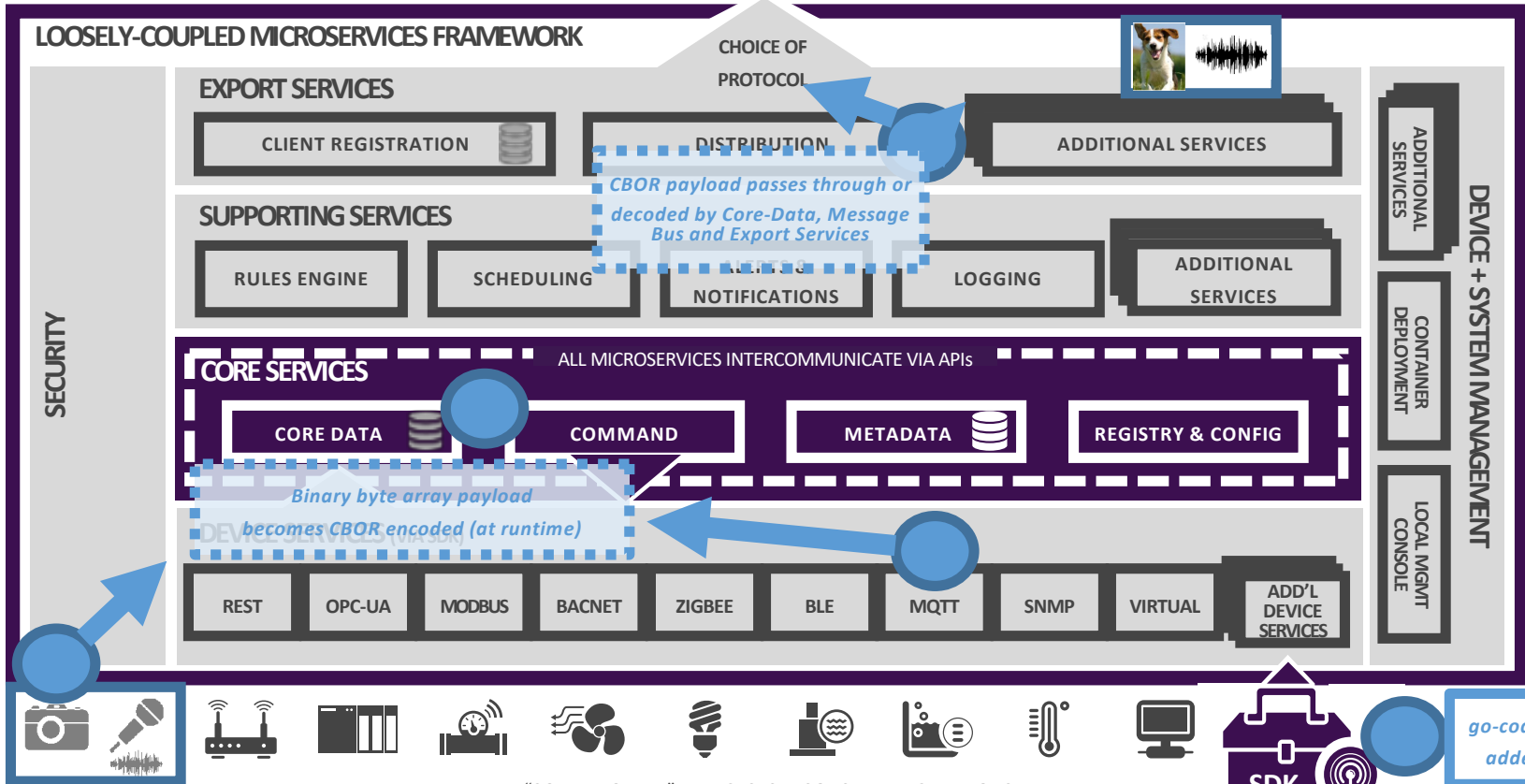
Tony Handle via reflection?

Fuji



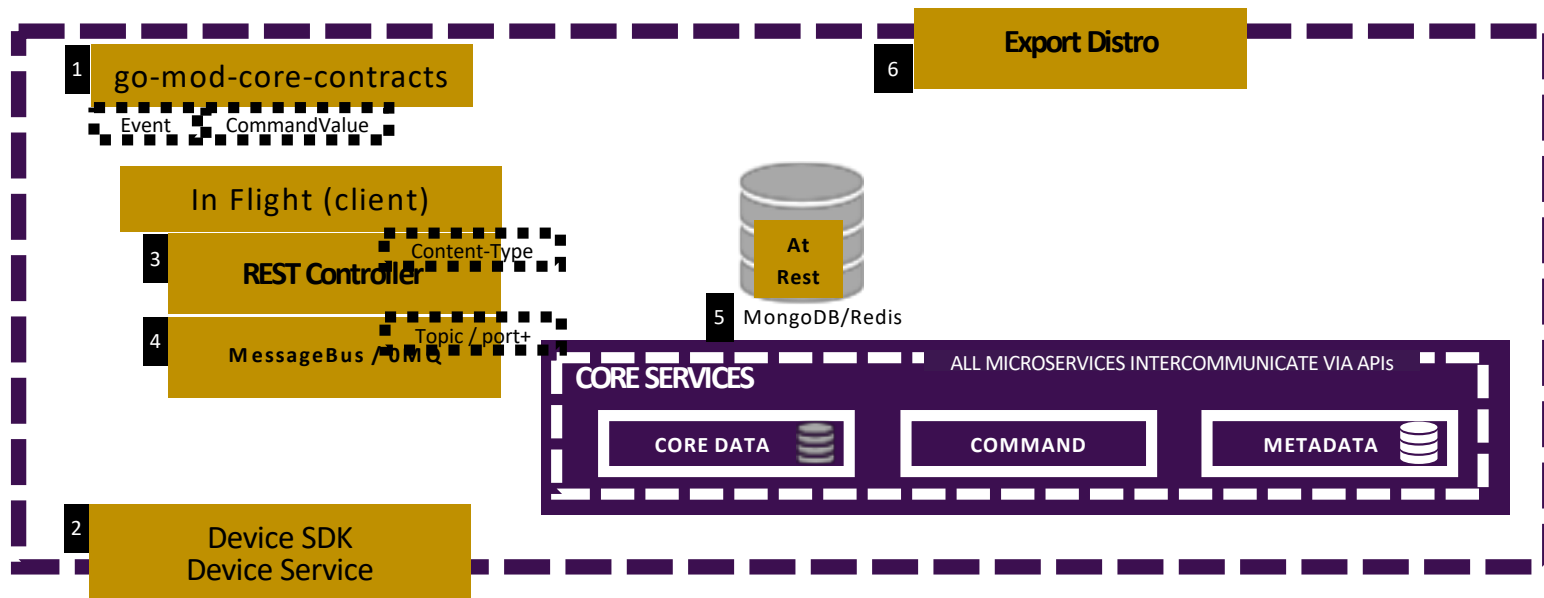
REQUIRED INTEROPERABILITY FOUNDATION
REPLACEABLE REFERENCE SERVICES

“NORTHBOUND” INFRASTRUCTURE AND APPLICATIONS



“SOUTHBOUND” DEVICES, SENSORS AND ACTUATORS

EdgeX Functionality / Data States Involved



1. Data model definitions and client accessors.
2. Device SDK data origination from "south".
3. New REST content-type to/from EdgeX Core-Data.
4. New messaging topic for binary event data (pass through).
5. Storage of events converted to BSON?
6. Optional transformation as events exit EdgeX to "north".

Preliminary Task Assignment & Roadmap

Feature	Milestone 1	Milestone 2	Edinburgh	Beyond
Contracts (Trevor)	<ul style="list-style-type: none"> Go: Update ReadingClient and model defs for Reading, Event(?), Filter(?) Go: Review PropertyValue and ValueDescriptor 			
Device SDK (Toby)	<ul style="list-style-type: none"> Go: Encode payload to new Reading.binValue field. Update command handlers, rest controller, processAsyncResult, CommandValueToReading helper. Go: Update device-simple to demonstrate binary encoded readings. Trigger through AutoEvents. Cloud removed Schedule and ScheduleEvents. Go: Update restController to pass content-type header as appropriate. 	<ul style="list-style-type: none"> C: Encode binary values into new readings field. C: Parity with Go SDK? Go: Add tests to assure coverage and validate handling of binary payloads. 	TBD	TBD
Device Service (Toby)	<ul style="list-style-type: none"> Go: <i>(completed)</i> Update device-mqtt-go to Edinburgh SDK 	<ul style="list-style-type: none"> Go: Add GetImage command to device-camera-go; populate reading.binValue 		<ul style="list-style-type: none"> Go: Add Media Mgmt Service
Core Data (Trevor)	<ul style="list-style-type: none"> Go: Decode and validate binary readings. Go: Store binary readings to file system. 	<ul style="list-style-type: none"> Go: Store binary readings < 16MB into db (mongoDB or Redis first?). Go: Add config driven option to reference file in stored db model. Go: Receive events with CBOR encoded readings (content-type: application/cbor) Go: Publish event as CBOR to MessageBus abstraction 		
Export Distribution (Janko)		<ul style="list-style-type: none"> Go: Send CBOR payload to MessageBus abstraction 		
Support Scheduler	<ul style="list-style-type: none"> Ready? Changes needed to complete this milestone? 	<ul style="list-style-type: none"> Go: Pass through to Omq bus/port 		<ul style="list-style-type: none"> Transformation service converts

MVP CBOR/Binary Encoding Scope / Decisions / Opens

Overall Encoding Guidelines/Assumptions

- Float values continue to offer Base64 encoding. Yes
- MVP 1.0 - Add DS SDK support for device services to add binary readings into a new binValue byte array field. When the Reading is serialized using CBOR, the bytes contained in the new byte array get efficiently encoded with no extra base64 encoding required...
- Time permitting; if we did not already review this:
 - <https://github.com/edgexfoundry/device-sdk-go/compare/master...tobiasmo1:cbor-initial?expand=1#diff-a2c6736ced964ce680cce6159abd2598>
- Should use of CBOR encoding trigger a new Correlation ID? .

OPENS

- **Developer Experience (DX)** requirements/objectives?
- Documentation GoDoc/other deliverables related to CBOR?
- Accept a new **binary resource** object in device profile; e.g., configuration.toml ingests from file reference?
- What is the influence of contracts event on MVP?
<https://github.com/edgexfoundry/go-mod-core-contracts/blob/master/models/event.go>



Process for Contributing EdgeX Source Code

- Fork the repositories you plan to work with and then pull your fork versus the EdgeX repositories directly.
- Details and coding guidelines found here:
 - <https://wiki.edgexfoundry.org/display/FA/Committing+Code+Guidelines>

EdgeX Developer Setup

- Implemented in ~~Go 1.10~~ **Go 1.11**
- Uses ~~Go Module Manager~~ Go Modules (vgo)
~~curl https://glide.sh/get | sh~~
- **Zero MQ** used for core-data, export-distro and rules engine.
~/dev/setup_zeromq.sh
- *GoLand, load edgex-go project*
 - ~~make prepare~~
 - *make build*

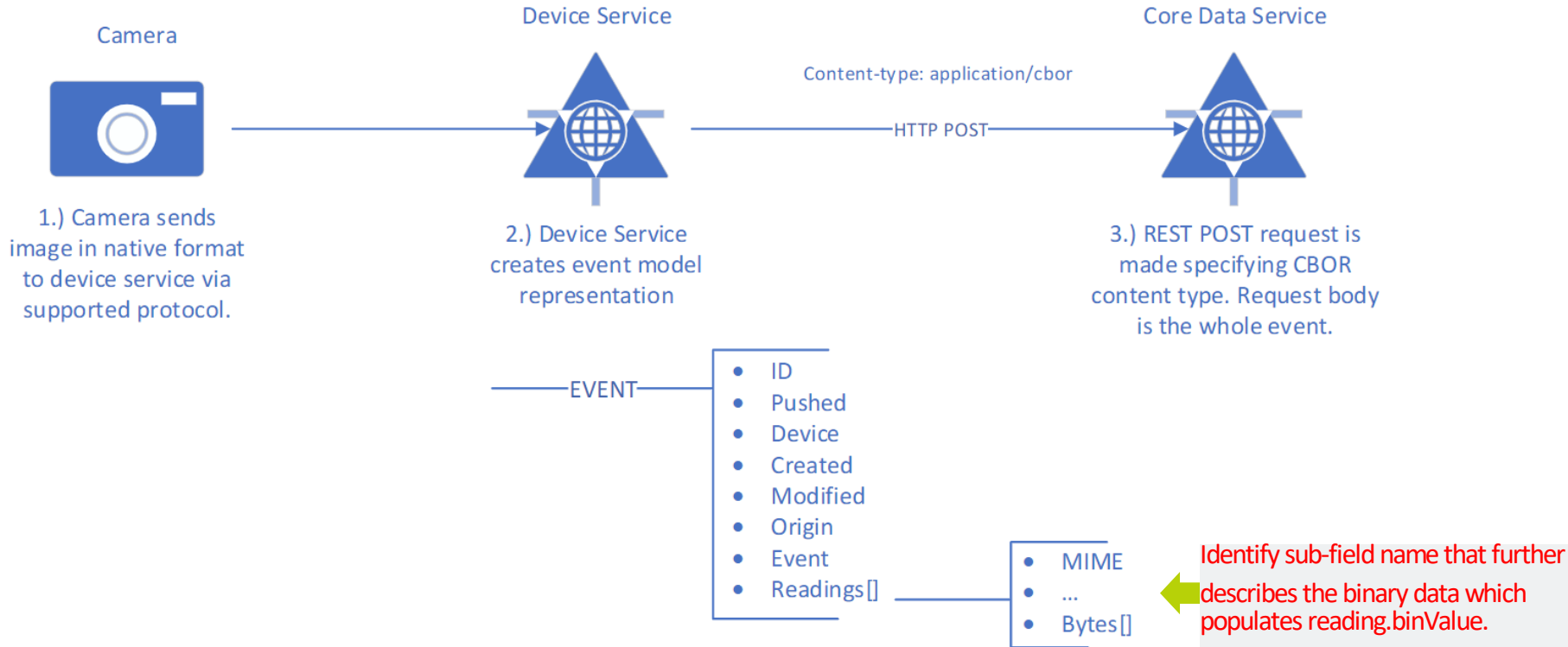
EdgeX Microservices Summary

Container/Service	Implemented in	Purpose
volume	N/A	Exposes 4 docker volumes . db-data, log-data, consul-config, consul-data
consul	Go	<ul style="list-style-type: none">The consul client Core service configuration for registration of an EdgeX microservice, its healthcheck, and to pull key/value pairs from consul service.Each microservice must register via consul client's ConsulInit.Clients available to connect to Consul agent implemented in Python, Java, PHP, Scala, Erlang/OTP, Ruby, NodeJS, C#
config-seed	Go	Populates the configuration/registry management microservice (consul). Note: config-seed and consul now share a common container. edgex-go is/becoming a mono repository (all inclusive) for accessing EdgeX microservices and client access libraries. I notice that config-seed is a straggler located in separate repo: https://github.com/edgexfoundry/core-config-seed-go
mongo	N/A	Scripts init of mongo 3.4.9 base image
core-command	Go	Provider for command(s) associated with device
core-data	Go	The core-data <i>client</i> sends events w/readings to this microservice via REST.
core-metadata	Go	The core-metadata client sends device services, devices, profile templates, and commands to this microservice via REST.
export-client	Go	Required for registration of cloud connectors and other export paths, CloudMQTT, Modbus, SNMP, etc.
export-distro	Go	https://github.com/edgexfoundry/export-go now lives in the mono repo: https://github.com/edgexfoundry/edgex-go/tree/master/export/distro
support-logging	Go	
support-notifications	Java (w/ Go client)	
support-scheduler	Java	
support-rulesengine	Java	

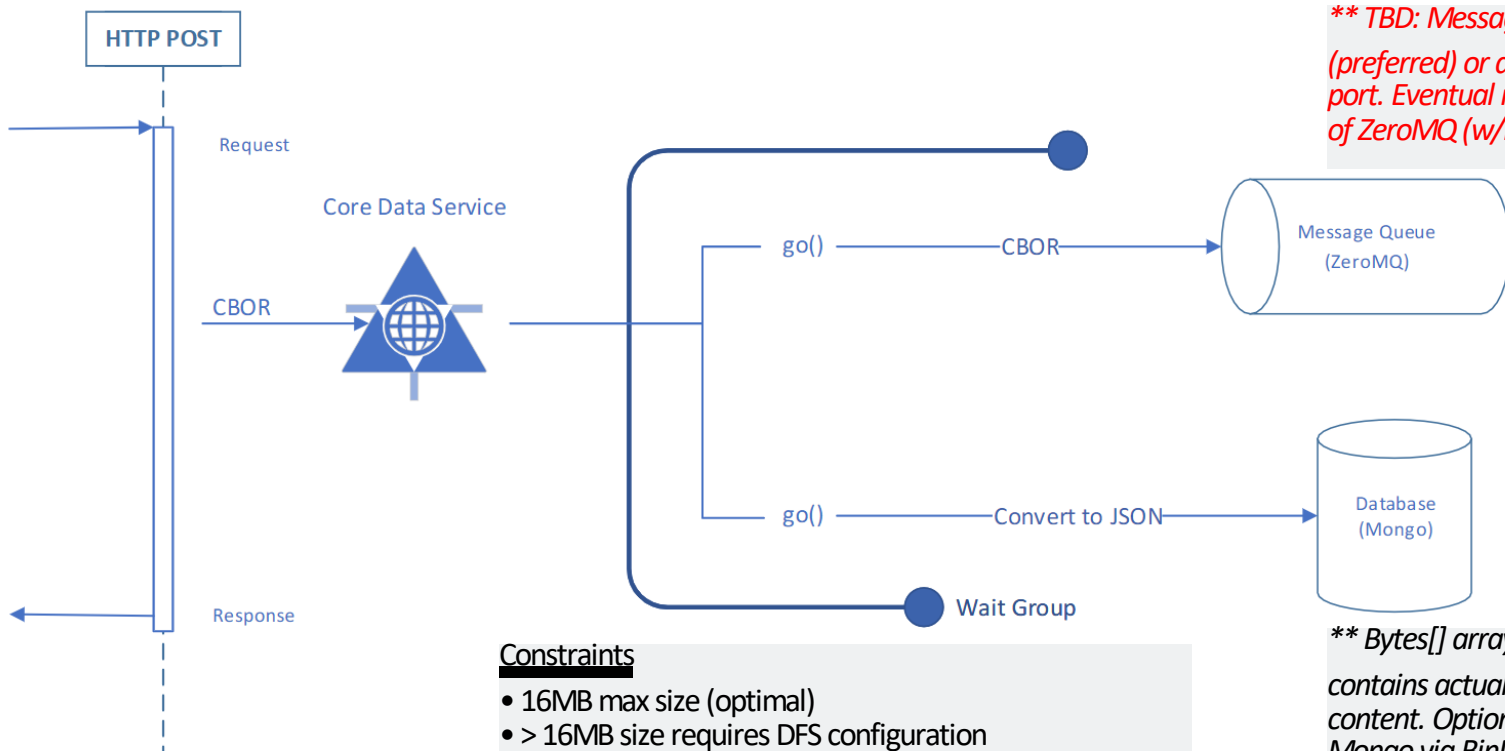
Add'l EdgeX Microservices

Container/Service	Implemented in	Purpose
edgex-device-virtual edgex-device-bluetooth edgex-device-snmp edgex-device-modbus edgex-device-fischertechnik edgex-device-bacnet edgex-device-mqtt		

Binary Data Ingestion via Core-Data



Core Data Handling of Binary Message



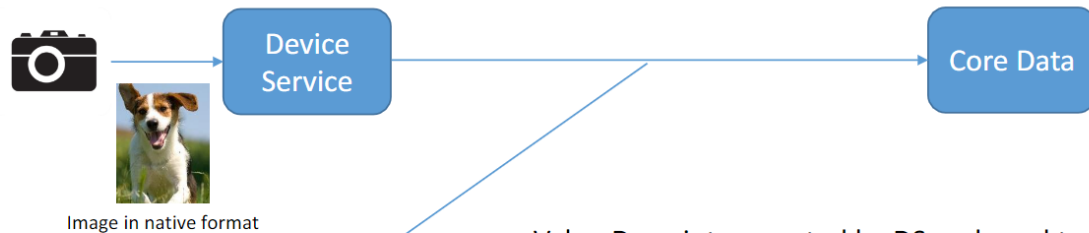
**** TBD: Message header (preferred) or additional port. Eventual replacement of ZeroMQ (w/MQTT?) ****

Constraints

- 16MB max size (optimal)
- > 16MB size requires DFS configuration
- Option to store images outside of Mongo on the file system and instead store the path to the image w/document

**** Bytes[] array on payload contains actual binary content. Option to support in Mongo via BinData type ****

Example 1 – Camera Image



Value Descriptor created by DS and used to identify payload of value. Optional ContentType property of a Value Descriptor would describe the format of the byte[] when the value is a byte[] (MPG, JPEG, GIF, MOV, ...)

```
Event  
ID: 0  
Pushed: 0  
Device: camera123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Readings
```

```
Reading  
ID: 0  
Pushed: 0  
Device: camera123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: JPEG  
Value:  
49424d434f52504f5241544  
94f4e32303036...
```

```
Reading  
ID: 0  
Pushed: 0  
Device: camera123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: JPEG  
Value: 786e124f0125a...
```

```
Reading  
ID: 0  
Pushed: 0  
Device: camera123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: JPEG  
Value: 1234b43c234c...
```

...

Note: the Value property was always a string and now can be a byte[]

Type should be that reflected in device profile: Float, String, Integer, ... and now Binary

“Chunks” of the image in CBOR encoded byte array with none exceeding 16MB size

Example 2 – Audio Content



Device Service

Core Data

```
Event  
ID: 0  
Pushed: 0  
Device: microphone123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Readings
```

Value Descriptor created by DS and used to identify payload of value.
Optional ContentType property of a Value Descriptor would describe the format of the byte[] when the value is a byte[] (MPG, JPEG, GIF, MOV, ...)

```
Reading  
ID: 0  
Pushed: 0  
Device: microphone123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: mp4  
Value:  
49424d434f52504f5241544  
94f4e32303036...
```

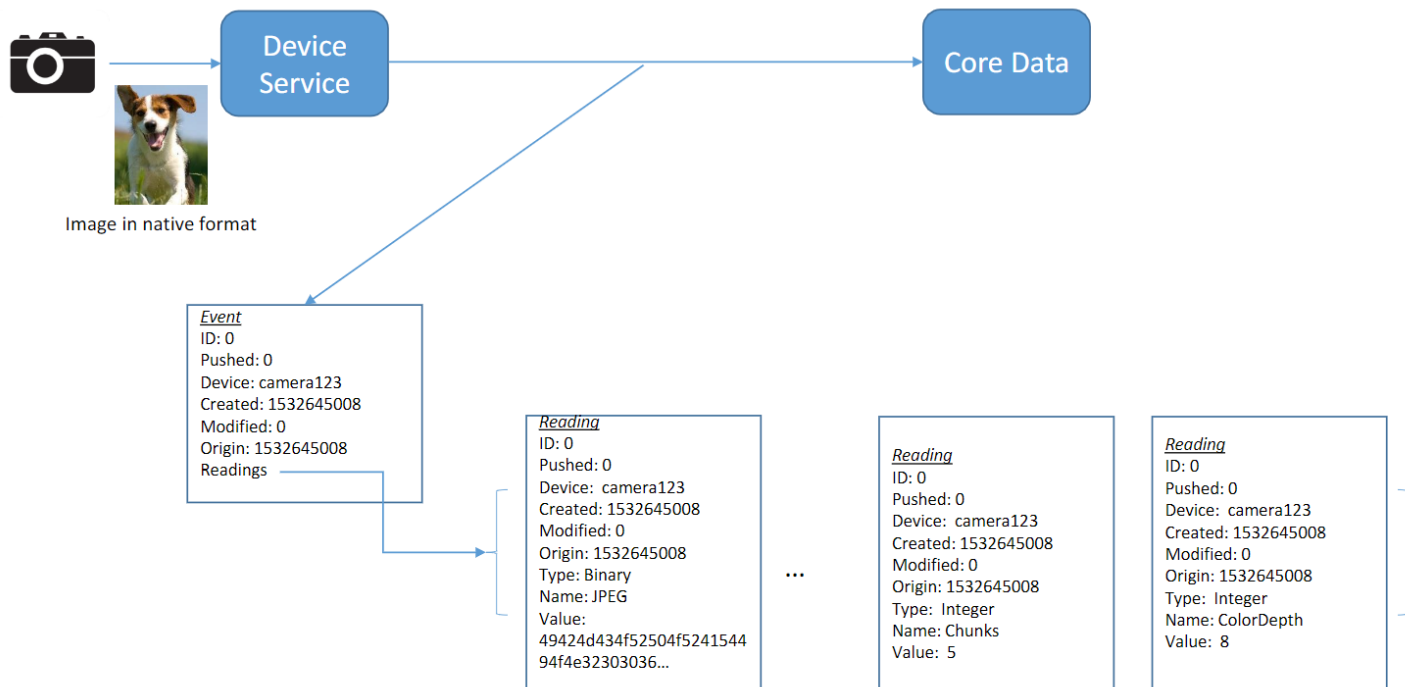
```
Reading  
ID: 0  
Pushed: 0  
Device: microphone123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: mp4  
Value: 786e124f0125a...
```

```
Reading  
ID: 0  
Pushed: 0  
Device: microphone123  
Created: 1532645008  
Modified: 0  
Origin: 1532645008  
Type: Binary  
Name: mp4  
Value: 1234b43c234c...
```

...

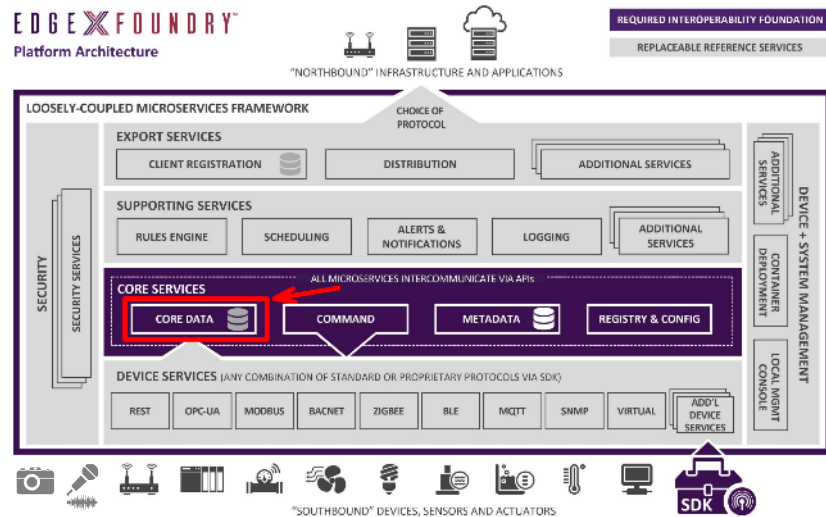
“Chunks” of the audio in CBOR encoded byte array with none exceeding 16MB size

Example 3 – Inference labels or metadata for VA



Review: Core Data Storage

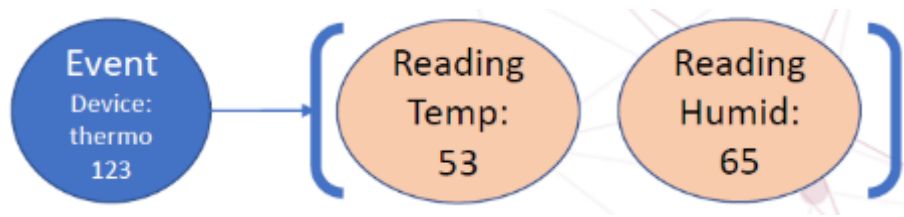
- Centralized persistence for ingested data.
- Device services collect data originating from “southbound” devices
- Calls on Core Data APIs for local persistence on edge system (gateway)
- Data stored until... it is processed for edge actuation and/or moved "north" and exported to Enterprise/Cloud
- Core Data provides APIs that other services can use to access the historical data.
- MongoDB actively in use while they explore other DBs



Review: EdgeX Data Types

Events & Readings

- Events are collections consisting of 1+ Readings
 - Associated to a device
- Readings represents sensed value (with optional origin)
 - Simple Key/Value pair holding sensed value(s)
 - Ex: Temperature: 72



Review: EdgeX Value Descriptors

The descriptor (key) in an EdgeX Reading's key/value pairs provides:

- Unique name
- Context and unit of measure
- Range validity rules
- Association labels / tags

```
{"name": "temperature",  
"description": "ambient temperature in Celsius",  
"min": -25,  
"max": 125,  
"type": "I",  
"uomLabel": "C",  
"defaultValue": 25,  
"formatting": "%s",  
"labels": ["room", "temp"]}
```

Core Data APIs – Readings

RESTful Value Descriptor APIs:

- POST, PUT, DELETE – Each assures descriptor is not already associated w/ existing Reading
api/v1/valuedescriptor
- GET - queries for value descriptor by:

Query By	Endpoint
Descriptor ID	api/v1/valuedescriptor/{id}
Name	api/v1/valuedescriptor/name/{name}
Label	api/v1/valuedescriptor/label/{label}
Device Name	api/v1/valuedescriptor/devicename/{device}
Device ID	api/v1/valuedescriptor/deviceid/{id}
Unit of Measure	api/v1/valuedescriptor/uomlabel/{uomLabel}

Core Data APIs - Events

RESTful Event APIs:

- POST - stores new Event w/ Readings
- PUT - updates entry (not often useful)
- DELETE - for data clean up (retention service)
`api/v1/event/device/{deviceId}`
`api/v1/event/scrub` (nuke all)
- GET - queries for Events by:

Query By	Endpoint
Event ID	<code>api/v1/event/{id}</code>
Device ID	<code>api/v1/event/device/{deviceId}/{limit}</code>
Timestamp	<code>api/v1/event/{start}/{end}/{limit}</code>
Readings by Device	<code>api/v1/event/device/{deviceId}/valuedescriptor/{valueDescriptor}/{limit}</code>

EdgeX – Administrative UI

localhost:4000

EdgeX Foundry Console 0.2.0

- Gateway
- Devices
 - Device
 - Device Service
 - Device Profile
- Multimedia
- Export
- Rule Engine

← Back ✓ Submit

New Device

Name: Description:

Admin State: Operating State:

New Device Profile

Device Profile:

Please upload a profile file.

New Device Addressable

Name: Method:

Protocol: Address:

Port: Path:

Publisher: User:

Password: Topic:

New Device Related Service

Device Service: