Toulouse 2018 - Agenda

- Berlin Namespace - Integration Update
    - Eclipse Kura+Hono+Kapua - Architecture
    - Demo
        - D2C: Telemetry / Events
    - Feature Branches
    - Issues

- Berlin Namespace - Integration Next Steps
    - P2C: Tenant / Authentication / Registry
    - Open Issue: library reuse

- Berlin Namespace - New API
    - P2C: Connection Events
    - C2D: Command and Control

- Integration Roadmap:
    - Name for Berlin Namespace => Eclipse IoT Namespace (beyond Hono)
    - Integration Milestones
    - Roadmap to Eclipse IoT Suite
    - Marketing and EclipseCON EU plan

IoT Messaging Interactions [No-Updates]

Devices - Devices can connect to the Cloud and have an identifier.
Devices can be smart devices (gateways) or simple constrained devices.

Interaction Type	Description	Notes
Device-to-Cloud (D2C)	Unsolicited messages sent by devices to the remote application (including Kapua). The conversation can be send-and-forget or request-reply.	
Cloud-to-Device (C2D)	The remote application initiates a conversation with a target device. The conversation can be a send-and-forget or request-reply.	
Device-to-Device (D2D)	The device initiates a conversation with another target device. The messages are initiated from a device and targeted to another device; messages will follow a device-to-cloud-to-device. The conversation can be a send-and-forget or request-reply.	
Platform-to-Cloud (P2C)	Message produced by the messaging platform to the application. E.g. Connection monitoring and life-cycle messages.	

Discussion resolution: do not focus D2D.

Device-to-Cloud (D2C) Messaging [No-Updates]

Unsolicited Messages Concepts

Message Type	Description	QoS Assumption	Volume Assumption	Priority Assumption	Persistence Assumption
Telemetry	A raw data reading - e.g. the value of a modbus register.	Low (At most once) High (At least once)	High	Low	No
Events	A data reading has reached a threshold and that triggered an event. A new configuration has been detected or applied.	High (At least once)	Medium	High	Yes
Alerts	Something happened and the device may not operate properly	High (At least once)	Low	Very High	Yes

Decision:

- QoS exactly once will not be supported

Eclipse IoT MQTT Namespace - Unified Kura, Kapua Hono Messaging Namespace

Eclipse IoT MQTT Namespace Proposal

Message Type	**Direc-tion**	**MQTT 3.1.1 Topic**	**MQTT QoS**	**MQTT Payload**	**Hono AMQP Address**	**Payload**	**Hono Message Persistence**
Telemetry	D2C	No auth: ● t/tenant/device/<sem-topic> Auth: ● t/tenant/device/<sem-topic> ● t/	0, 1	Opaque	telemetry/tenant/ ApplicationProperties: device_id, orig_address AMQP Message	Opaque - format can be specified at the device-level from Registry	No
Events	D2C	No auth: ● e/tenant/device/<sem-topic> Auth: ● e/tenant/device/<sem-topic> ● e/	1	Opaque	event/tenant/ ApplicationProperties: device_id, orig_address AMQP Message	Opaque - format can be specified at the device-level from Registry	Yes
Alerts	D2C	No auth: ● a/tenant/device/<sem-topic> Auth: ● a/tenant/device/<sem-topic> ● a/	1	Opaque	alert/tenant/ ApplicationProperties: device_id, orig_address AMQP Message	Opaque - format can be specified at the device-level from Registry	Yes
Connection Events	P2C	-	-	-	event/tenant/ ~~platform/tenant~~ ApplicationProperties have no orig_address property set	Defined by Hono	Yes, by conf.
Device Subscription	C2D	Noauth: c/tenant/device/req/# Auth: c/req/#	0		-	See notes below	-

Request Topics	C2D	c/req/cmd/reqID	0	Opaque	control/req/cmd/reqId		No
Response Topics	D2C	Noauth: c/tenant/device/res/reqID/status Auth: c/res/reqID/status	0	Opaque	control/res/tenant/device/reqId/status	CorrID in header	No
Device to Device	D2D	While possible with the MQTT namespace above, we may not want to enable it for security reasons.					

Notes:

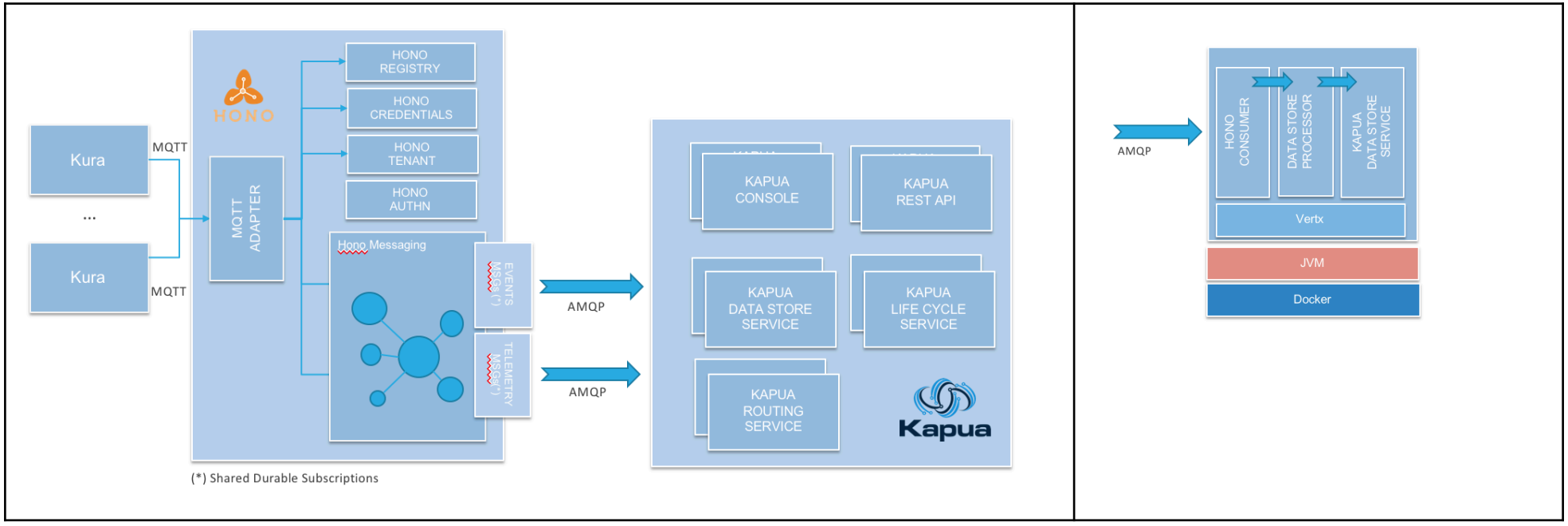
- Can the Messaging Infrastructure add device-specific message headers into the messages? For example, data vs control message?
- Possible examples are: originating protocol, content-type per message type, …
- Options for request topics:

1. ~~**control/req/CONF/1234567/GET/resource.id**~~
2. **control/req/1234567/CONF/GET/resource.id**
3. **control/req/CONF-GET-resource.id/1234567**
4. ~~**control/req/CONF/1234567?mode=GET&id=resource.id**~~
5. **control/req/CONF%2FGET%2Fresource.id/1234567**
6. ~~**control/req/CONF/1234567**~~
   ~~**{“mode”:”GET”, “id”:”resource.id”}**~~

Eclipse IoT MQTT Namespace - Toulouse 2018 Demo

Step 1 - Architecture Diagram

The demo implements the device-to-cloud interaction for telemetry messages. Incoming telemetry messages are sent to Kapua Data Store service where they are persisted..



Links

- https://github.com/eclipse/kapua/tree/feature-eclipseiot
- https://github.com/eclipse/kura/wiki/Generic-cloud-services

Next steps

- Integration with Kapua device lifecycle service

Service Integration - Device Registry [No-Updates]

Tenant API

API	AMQP 1.0 Endpoint	Response	Notes
Get Tenant Registration	tenant/,get,tenantId	Tenant status and configuration for protocol adapters.	Post authentication

Device API

API	AMQP 1.0 Endpoint	Response	Notes
Assert Device Registration	registration/tenant,assert,deviceId[,gatewayid]	Signed JSON Web Token with device's tenant and validity period. Can also return default properties of the device.	Post authentication and connection
Get Registration Information	registration/tenant,get,deviceId		

Device Model

- Tenant Id
- Device Id
- Optional name/value pair default properties

Base Implementation

- https://github.com/eclipse/hono/tree/master/service-base
- https://github.com/eclipse/hono/tree/master/service-base/src/main/java/org/eclipse/hono/service/registration

Service Integration - Credentials (D2C) [No-Updates]

Credentials API - Used in the Device-to-Cloud (D2C)

API	AMQP 1.0 Endpoint	Response	Notes

Get Credentials	credentials/tenant/,get,"hashed-password",auth-id[,client-id]
	
	Hono-TODO: add connection property bag with "client-id" as a well-known property in that set.
		<pre>{
		  "device-id": "4711",
		  "type": "hashed-password",
		  "auth-id": "sensor1",
		  "enabled": true,
		  "secrets": [{
		  "not-after":
		"2017-12-24T19:00:00+0100",
		  "pwd-hash":
		"AQIDBAUGBwg=",
		  "salt": "Mq7wFw==",
		  "hash-function": "sha-512"
		  }]
		}</pre>
			Pre authentication

MQTT Authorization ACLs (Hono-TODO):

- can publish under: [t,e,a,r]/tenant/#
- If not gateway, can publish under: [t,e,a,r]/tenant/device-id/#
- If gateway, can publish [t,e,a,r]/tenant/child-devices/#
- Can subscribe only to: c/tenant/device-id/#

Other MQTT Protocol Adapter Assumptions:

- clean-session MUST be true. Connect requests with clean-session=false will be rejected.
- retain flag MAY be true. Messages will be confirmed if necessary, and an "x-opt-retain" application property will be added.
- Messages published with qos=2 are not acknowledged, discarded and tracked in malformed-messages Hono metric.
- LWT will be ignored if provided. Hono will log it and track it in "some" Hono metric.

Service Integration - Authentication (P2C) [No-Updates]

Authentication API - Used in the Platform-to-Cloud (P2C)

Decision: For the purpose of Hono + Kapua integration, support for the Hono Authorization API in the Kapua Provider will be addressed in a Phase 2.

Service Integration - Connectivity API [No-Updates]

Hono-TODO: Add Connectivity API for life-cycle messages

Notifications on connection established are guaranteed.

Notifications on connection losses are best effort. (Hono-TODO: is there a way to guarantee those notifications as well?)

EclipseCON EU 2018 - Tentative Plan

- Synchronized releases - Kura, Hono, Kapua, Paho
- All docker images on Eclipse Org Docker Hub
- Getting Started Guide
    - Setup guide
    - Kura applications bundles with generated traffic
    - Data visualization of some sort (grafana or InfluxDB chronograf)
- New project for Eclipse IoT Integration?
- Hosted instance of the integrated suite?
- Published Eclipse IoT MQTT Namespace (conventions? guidelines?)
    - Communication of adoption of the MQTT Namespace by commercial products (Eurotech, Bosch, …)
- Testbed
    - Migration of one of the existing testbed to the new Namespace?
    - Add XDK support into the asset tracking testbed?
- Plan next integrations
    - Ditto
    - MQTT v5

References [No-Updates]

MQTT

https://esf.eurotech.com/docs/mqtt-namespace
https://www.eclipse.org/hono/user-guide/mqtt-adapter/
http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support
https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html

Digital Twin

https://esf.eurotech.com/docs/kura-wires-mqtt-namespace
https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins
<Eclipse Ditto>

Open Questions [No-Updates]

- Identity of connected devices, gateways and assets. Common picture and glossary.
- How to propagate asset identities through device identities through Hono
- Hono Federation
- Multicast to Device Groups
- Assert Credentials API (case: authentication through GitHub, Google, etc)
- Do we need a prefix for the whole Eclipse IoT MQTT namespace?

Berlin Namespace - Digital Twin Namespaces [No-Updates]

Hono Device Registry

- Each device has an ID
- One device can be registered “via” another device

Kapua+Hono

- Kapua will registry top-level devices only. Kapua Assets will not be registered in the Hono Device Registry