Eclipse Hono and all things IoT messaging

IoT Day Grenoble 2018

Dejan Bosanac, Red Hat
Who am I

Dejan Bosanac

Software Engineer @ Red Hat
- Messaging and IoT

Open source committer
- Eclipse Hono
- Eclipse Kapua
- Apache ActiveMQ
Eclipse Hono provides a **uniform API** for interacting with **millions of devices** connected to the cloud via **arbitrary protocols**.
Eclipse Hono


- An Eclipse Foundation IoT project ...
  - Bosch and Red Hat as main contributors
- [https://www.eclipse.org/hono/](https://www.eclipse.org/hono/)
Eclipse Hono

- Open source IoT connectivity platform running on …
  - Kubernetes
  - OpenShift
  - Docker Swarm
- On-premise & in the cloud
- Provided by a set of Docker containers
Eclipse Hono

Goals

- Tailored general messaging for IoT solutions
- Provide standard APIs for interacting with devices
- Support for arbitrary protocols (MQTT, AMQP 1.0, HTTP, …)
- Support different underlying messaging infrastructures
  - AMQP 1.0 based
  - JMS
  - Apache Kafka
  - RabbitMQ
Eclipse Hono

Features

- Scalability
- Multi-tenancy
- Device-based security
- Multi-protocol support
Cloud

optimized for throughput
scale-out with messages

Telemetry

optimized for reliability
scale-out with devices

Command & Control

many existing protocols
HTTP, MQTT, CoAP etc

arbitrary providers &
deployment options

Things
Building Blocks

Devices — Protocol Adapters — API Endpoints & Security — AMQP messaging network

end to end flow control

HTTP
MQTT etc.

AMQP 1.0

Business Applications

AMQP 1.0
Micro-Service Design
≥ 0.5-M7

MQTT Devices

MQTT Adapter

Telemetry

Event

Hono Messaging

Authentication

Auth Server

Device Registration

Device Registry

Credentials

ActiveMQ Artemis Broker

Qpid Dispatch Router

Business Applications

Telemetry

Event

Provided by 3rd Party/Demo Implementation
Eclipse Hono

Telemetry & Event

- used by devices to **send data/event downstream**
- leverages on “**direct messaging**” …
  - Telemetry
  - Devices can send data only if consumers are online
  - No broker involved
- … “**store and forward**”
  - Event
  - Broker for storing event with a “ttl” eventually
- consumers receive data published by devices belonging to a particular tenant
Eclipse Hono

Device Registration

- used to make Hono **aware of devices** that will connect to the service
- solutions/consumers may use the API to get information about devices

- operations
  - register, deregister, get information, assertion

- for every message sent by a device …
  - a **registration assertion** (JWT) is attached by the protocol adapter
  - it’s verified by messaging before sending the message downstream
  - a disabled device will have such check fails
Registration Assertion

1. publish telemetry
2. assert (tenant-id, device-id)
3. check if device is registered and enabled
4. Json Web Token
5. create message (incl. token)
6. forward message
7. validate token
8. forward message
Eclipse Hono

Credentials

- handle authentication for devices on protocol adapters
- used by protocol adapters to retrieve credentials used to authenticate devices connecting to the adapter (MQTT, HTTP, ...)
- different types of credentials
  - psk, hashed password, public key, ...
- operations
  - add, get, update, remove
- Where an identity management system is already in place (i.e. Keycloak) ...
  - needs for having a “facade” from this API to such a system
Eclipse Hono

Authentication

- handle authentication between components (protocol adapters, messaging, ...)
- used by clients/components for getting a **token** asserting ...
  - subject’s identity
  - granted authorities
- other services will use such a token to make authorization decisions on a client’s request to read or write from/to a resource or to invoke a certain operation
  - i.e. messaging checks if an adapter can write telemetry data
- Where an **identity management system** is already in place (i.e. Keycloak) …
  - needs for having a “facade” from this API to such a system
Features Hono 0.5

- Uniform APIs for consuming telemetry data and events
- MQTT, HTTP protocol adapters
- Device-level Authentication
- Tenant based Security Model
- Horizontal Scalability
Future

● Performance and scalability testing and tuning
● Continue improving OpenShift and EnMasse integrations
● Command and control API
● Tenant API
Eclipse Hono
Command & Control

- used by applications to **send commands to devices**
- command execution can be “just in time” or “deferred”
  - **just in time**: command already executed, the response from device contains the result
  - **deferred**: command not executed yet, the response from device specifies it’s accepted; for long running operations the result will be provided later
Simple deployment

HTTP, MQTT

AMQP 1.0

API Endpoints & Security

Qpid Dispatch Router

ActiveMQ Artemis Broker

AMQP 1.0
Routing vs Brokering

Producer

Broker

Consumer

Send message

Accepted

Send message

Accepted
Routing vs Brokering

Producer

Router

Consumer

Send message

Accepted

Send message

Accepted
Addressing semantics

- Store and Forward
  - Queue
  - Topic

- Direct
  - Anycast
  - Multicast (Broadcast)
Scalable deployment

- EnMasse …
  - a messaging-as-a-service platform
  - elastic scaling
  - multiple communication patterns
  - more info: enmasse.io
  - … and more and more …
Basic idea
Messaging-as-a-Service

- Open source cloud messaging running on Kubernetes and OpenShift
- enmasse.io
Features

- Multiple communication patterns: request/response, publish/subscribe and competing consumers
- Support for “store and forward” and direct messaging mechanisms
- Scale and elasticity of message brokers
- AMQP 1.0 and MQTT support
- Simple setup, management and monitoring
- Multitenancy: manage multiple independent instances
- Deploy “on premise” or in the cloud
Address types

- Queue
  - store-and-forward = true
  - multicast = false

- Topic
  - store-and-forward = true
  - multicast = true

- Anycast
  - store-and-forward = false
  - multicast = false

- Broadcast
  - store-and-forward = false
  - multicast = true
Flavor examples

- Persistence
  - In memory
  - Persisted
- Scaling
  - Single broker
  - Pooled
- HA
Future
In progress/TODO

- Authentication and authorization
- Service broker API
- HTTP(S)
- Broker address space
  - Message grouping
  - Distributed transactions
  - Message ordering
- Multiple flavors
  - Apache Kafka?
- ...

Resources

- Eclipse Hono - https://www.eclipse.org/hono
- EnMasse - http://enmasse.io
- ActiveMQ Artemis - https://activemq.apache.org/artemis/
- Qpid Dispatch Router - http://qpid.apache.org/components/dispatch-router/
Thank you! Questions?