www.connectedthingslab.com

Version 0.1

VeliSphere

VeliSphere
Data Exchange Format (VDEF)

Built on Open Standards and Open Technologies

Connected Things Lab

Disclaimer

This presentation is the property of Thorsten Meudt (the owner) and is strictly confidential. It contains information intended only for the person to whom it is transmitted. With receipt of this information, recipient acknowledges and agrees that:

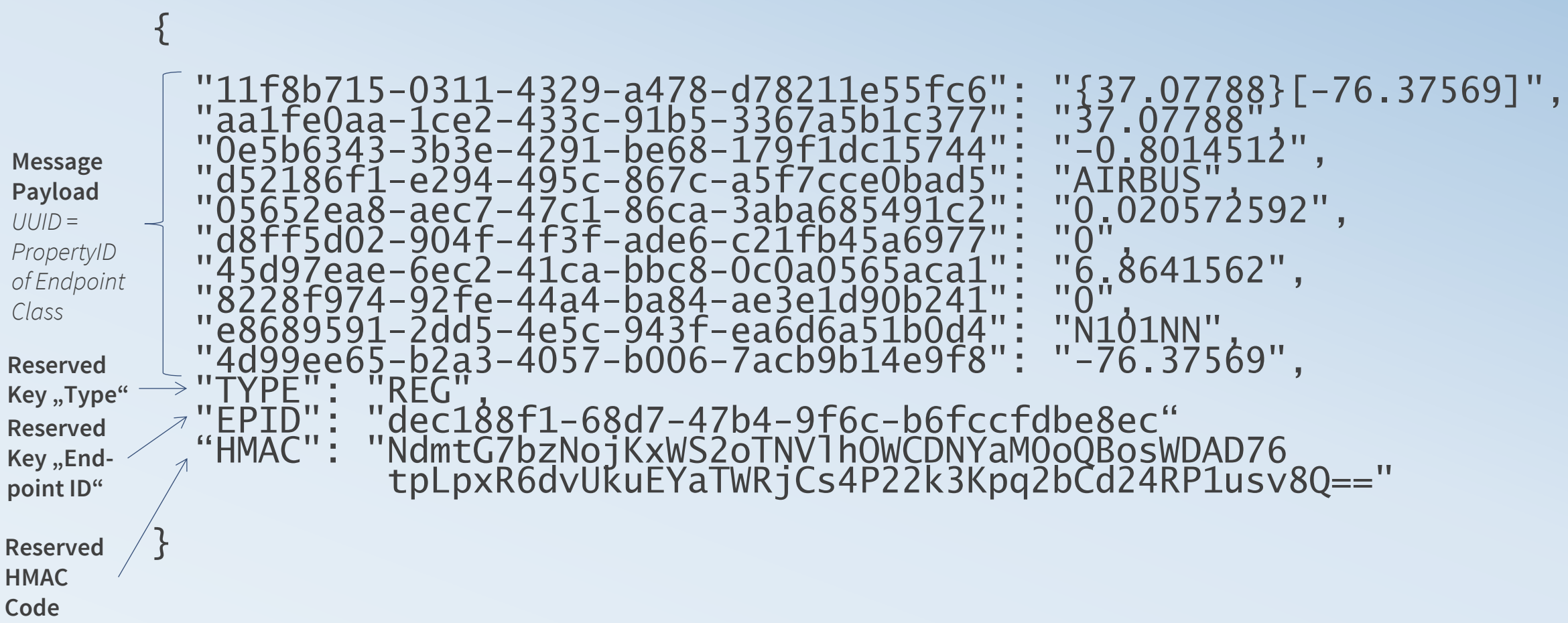
(i) this document is not intended to be distributed, and if distributed inadvertently, will be returned to the owner as soon as possible

(ii) the recipient will not copy, fax, reproduce, divulge, or distribute this confidential information, in whole or in part, without the express written consent of the owner

(iii) all of the information herein will be treated as confidential material with no less care than that afforded to its own confidential material.

Definition

- VDEF is a data packaging format rather than a protocol
- It follows the principle of “Key X” / “Value Y”
- The key is either
  - A reserved key identifying control functions (such as „getIsAlive“ or „getAllProperties“
  - A 16 byte UUID representing a property of the sending endpoint‘s property class
- Key and value are always to be encoded as strings, the backend will convert them to the appropriate data type as needed
- One or more key/value pairs form a message. The message is serialized into standard JSON.
- The VDEF JSON message forms the main communication piece that gets transported to/from the VeliSphere Controller
- VDEF can be sent over any VeliSphere-supported application layer protocol
- VDEF provides built-in authentication (HMAC SHA-1 based), for encryption it relies on the underlying protocol (HTTPS, AMQP-TLS)
- VDEF does not handle security in any way
  - Authentication/Authorization/Encryption (SSL) is handled on the application layer protocol level (AMQP, HTTP-ReST, MQTT will be supported)



VDEF JSON Message Example

Message Types

- REG — Regular Messages, Controller will run them through business logic engine
- CTL — Control Message, will executed defined actions as defined by the reserved keys. Will bypass BLE
- CNF — Configuration Message, contains key=configurable property of endpoint, value = config value. Directly sent to endpoint via Toucan Web Service. Unidirectional, only to be sent to Endpoint, not to Controller.

- For messages to an endpoint:
  - Key: getIsAlive
    - request endpoint to submit current state as value to controller
  - Key: getAllProperties
    - request endpoint to submit a message containing the value of all sensor properties
- For messages to the controller
  - Key: setState→ Sets Endpoint State. Possible values: see seperate list

Reserved Keys

Montana/Endpoint/Endpointstate:

- UNKNOWN
- REACHABLE
- CONFOK
- CRITICAL
- CONFOFF

Values for ENDPOINT STATE