

Device endpoint protocol description

1. Introduction

In this document is described how a client device can register and send data securely to a server device.

2. Device endpoint setup

To register a client device the following steps are involved

- If not already present, create an identity (e.g. account) to which you want to store data for (Server)
- Generate and store a secure token associated with the identity which will be used as a PSK (Server)
- Share the PSK with the client (Server -> Client)
- Install or generate a (X509) certificate unique to the device (Client)
- Send a POST request over a TLS connection with client authentication using the device's certificate with the PSK in a JSON body (Client -> Server)

Example request:

```
POST /device_endpoint HTTP/1.1
Content-Type: application/json
Content-Length: 23

{"PSK": "secret_token"}
```

- Validate the PSK (Server)

When the PSK validates

- Bind the certificate to the identity
- Respond with a 200 HTTP status code to notify the client that the registration succeeded

When the validation fails

- Respond with a 400 HTTP status code to notify the client that the registration failed

3. Sending data

When the certificate is successfully bound, data can be send to the server by the client device. This is done by sending a POST request with a multipart body over a TLS connection with client authentication using the bound certificate. The server identifies the client based on the certificate and thus knows to which identity to associate the data with.

Example request

```
POST /device_endpoint HTTP/1.1
Content-Type: multipart/form-data; boundary=-----
-9051914041544843365972754266

-----9051914041544843365972754266
Content-Disposition: form-data; name="hr_measurement";
filename="hr_measurement_2018_05_01"
Content-Type: text/plain

1526292559214,95
1526292559839,96
1526292560397,95
1526292560919,97
1526292561361,98
1526292561911,95
1526292562392,94

-----9051914041544843365972754266--
```

On a successful received request the server must respond with a 200 HTTP status code, on failure the server must respond with an appropriate HTTP status code.