remove-digital-twin

Service Description

Abstract
This document provides service description for the **remove-digital-twin** service.

Contents

1 Overview

This document describes the **remove-digital-twin** service, which makes it possible to remove a digital twin. The service will remove and deregister all services associated with the digital twin and remove the digital twin from the digital twin hub database.

The rest of this document is organized as follows. In Section 2, we describe the abstract message functions provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned functions.

1.1 How This Service Is Meant to Be Used

The system operator should use the **remove-digital-twin** service when it is no longer desired to have that digital twin system in the local cloud.

1.2 Important Delimitations

No delimitations.

1.3 Access policy

Available only for system operators, it is necessary that a sysop certificate is provided.

2 Service Interface

This section describes the interfaces to the service. The **remove-digital-twin** service is used to remove a digital twin system. Various parameters are representing the necessary system input information. In particular, each subsection names an interface, an input type and an output type, in that order. The input type is named inside parentheses, while the output type is preceded by a colon. Input and output types are only denoted when accepted or returned, respectively, by the interface in question. All abstract data types named in this section are defined in Section 3.

The following interfaces are available.

2.1 interface HTTP/TLS/URL (QueryParams) : Void

Profile type	Type	Version
Transfer protocol	HTTP	1.1
Data encryption	TLS	1.3
Encoding	URL	-
Method	DELETE	-

Table 1: HTTP/TLS/JSON communication details.

3 Information Model

Here, all data objects that can be part of the **remove-digital-twin** service provides to the hosting System are listed in alphabetic order. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.2, which are used to represent things like hashes and identifiers.

3.1 struct QueryParams

Field	Type	Mandatory	Description
address	Address	yes	Network address.
port	PortNumber	yes	Network port.

3.2 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Address	A string representation of the address
PortNumber	Decimal number in the range of 0-65535

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	4.6.1		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	4.6.1	