Historian Service SD

**Abstract**

This document defines the Historian Service functionality within Arrowhead generation 4.0. The Historian provides methods for storage, retrieval, export and filtering of sensor data. It also has features for storage of generic files such as text documents, images, reports etc.

One of the main design goals of the Historian is to provide storage capabilities for sensor and actuator devices, basic data analytics, and export for human operators.

1. Service Description Overview

The Historian Service provides two interfaces to both store and retrieve sensor data and/or files.

1. Abstract Interfaces

This Service provides four functionalities, ListSystems, ListServices (used to list active systems and services), Store and Fetch, that are used to either store sensor data (or upload files) to the Historian servicer or to retrieve sensor data or download files. The interfaces can optionally support a number of commonly used semantics formats in order to allow heterogenous systems to exchange information.

# ListSystems

This interface lists all systems that have inserted service data. The output is an array of system names.

# ListServices

This interface lists all services for a system that have inserted service data. The output is an array of service names.

# Store

The Store interface provides methods for storing sensor data as well as generic files. A provider can push sensor readings to the store interface. The Historian validates all incoming messages and returns an error if an incorrect message has ben pushed. If a supported format is used, then the Historian will decode the message and extract all data fields, time stamps etc.

# Fetch

The Fetch interface provides methods for fetching sensor data as well as generic files. The Fetch interface also provides methods for data export, where data can be aggregated and exported in different formats and semantics. The Fetch interfaces thus provides an abstraction interface to basic database queries.

The Retrieve interface can also take a number of optional parameters.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| count | Limits the number of values in the response | count=10 |
| sigX | Only includes signals with a specific name. Use sigX=name, where x goes from 0 to the number of requested signals | sig0=IndoorTemperature &sig1=OutdoorTemperature |
| sigXcount | Limits the number of values per individual signal | sig0=IndoorTemperature &sig0count=10 |
| from | Restricts data so that only entries with a timestamp greater or equal to is returned | from=1592666683 |
| to | Restricts data so that only entries with a timestamp smaller or equal is returned | to=1605562117 |

1. Abstract Information Model

A sensor data message contains, (at least), the following information, as presented in Table 1. This abstract model is not based on any standardized format. Instead there are many formats that can support this type of information.

# SystemList

|  |  |
| --- | --- |
| **Field** | **Description** |
| Systems | An array of systems that have active services |

Table 1 – SystemList

# ServiceList

|  |  |
| --- | --- |
| **Field** | **Description** |
| Services | An array of services that have stored data |

Table 2 – ServiceList

# DataMessage

|  |  |
| --- | --- |
| **Field** | **Description** |
| Arrowhead System | The System that generated or stored a measurement. |
| Base time | If set, this is the timestamp when a reading was taken. If not set, the input message was either a file or a reading taken approximately “now”. |

Table 3 – DataMessage

1. Non-functional Requirements

The Historian service must run on a sufficiently powerful host that provides enough storage and processing capacity for the number of clients that are using it. A database must also be installed and properly configured.

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2018-09-17 | G4.0 d1 | Initial | Jens Eliasson |
| 2 | 2018-10-30 | G4.0 d2 | Updated description | Jens Eliasson |
| 3 | 2019-03-20 | G4.0 | Updated data format and parameters | Jens Eliasson |
| 4 | 2020-05-27 | G4.1.3 | Updated with sigXcount | Jens Eliasson |
| 5 | 2020-11-17 | G4.1.3 | Updated with more messages | Jens Eliasson |
|  |  |  |  |  |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |
| 2 |  |  |  |