4subsea

DataReservoir.io™ Connector

DataReservoir.io™ Connector

- Functional overview
- Network and Connectivity
- System Requirements
- Deployment

Functional Overview

- Transfer data from customer sources into DataReservoir.io™
  - Continuous streaming for online scenarios
  - Data compression and value filtering for optimized bandwidth usage
  - Built-in resiliency to handle downtime and unreliable networks
  - Data always encrypted in transit using TLS (Azure IoT Hub reference)
- Support industry standard protocols OPC/UA
  - Stream historical data from OPC HDA and OPC UA HDA
  - Stream current data from OPC UA DA enabled sources
- Extendable
  - Can be extended to support custom OData and REST API sources

Network and Connectivity

Connectivity Requirements

- Connector to data source
  - OPC UA HDA – both binary (any port) or web service (HTTPS) are supported
  - OPC HDA – local COM or remote DCOM are supported

- Connector to DataReservoir.io™
  - HTTPS - outbound communication to Azure IoT Hub (port 443)
  - HTTPS - outbound communication to Azure Application Insights (port 443)

Firewall Requirements

- The following hosts, IPs and ports must be allowed from the site of the DataReservoir Connector to the Internet.
- IMPORTANT: to ensure stable and performant connectivity, it is required that communication should not pass through a web proxy

What	DNS	IP (if specific)	Port
Data Receiver	reservoir-iot-prod.azure-devices.net	40.113.176.174 (pre 2021.08.01 : 40.118.27.192)	443
ALM Services Service Health Telemetry	dc.services.visualstudio.com dc.applicationinsights.microsoft.com	40.114.241.141 104.45.136.42 40.84.189.107 168.63.242.221	443
ALM Services Service Health Telemetry	rt.services.visualstudio.com rt.applicationinsights.microsoft.com	23.96.28.38 13.92.40.198	443

System Requirements

System Requirements

DataReservoir.io™ Connector system requirements depend on the amount of data to be transferred. Use the following base line:

- Per <500 tags, low-frequency up to 10Hz data rate
  - 1x virtual machine in customer network hosting the connector
  - (If data sources does not support historical access, downtime in streaming is to be expected during OS, software maintenance or network interruptions)
- Machine specifications:
  - OS: Windows Server 2016 or 2019 with Microsoft .NET Framework 4.7.2
  - OPC Core Components version 3.00.108 (https://opcfoundation.org/developer-tools/samples-and-tools-classic/core-components/)
  - Hardware/virtual machine with 4 cores, 8GB memory
  - Reliable network connectivity, minimum two NICs for dedicated internal and external network traffic is recommended

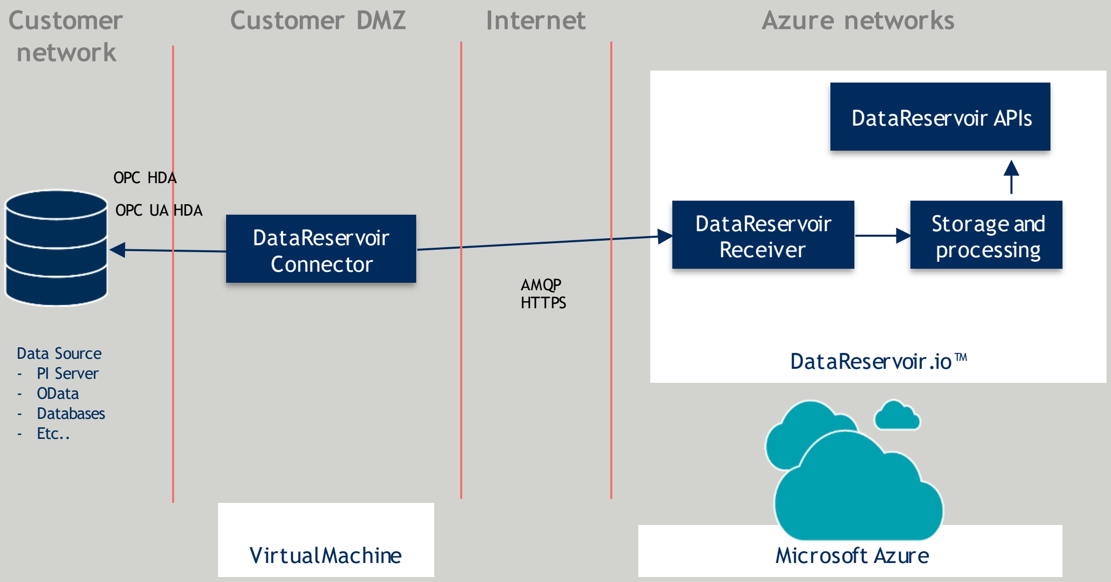
Deployment Topology

Connector Deployment

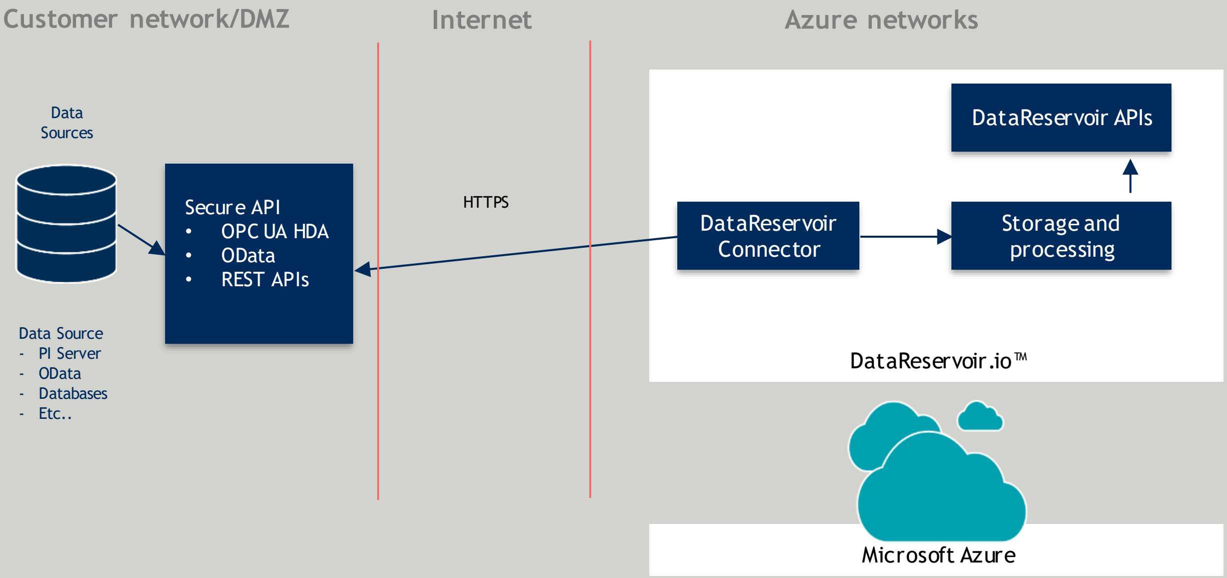
DataReservoir.io™ Connector must be deployed with outbound connectivity to data sources and DataReservoir.io™ endpoints.

The following illustrations are examples of topologies where Connector is deployed, either close to the data source (Edge), or running in the cloud.

Edge Deployment Topology



Cloud Deployment Topology



4subsea

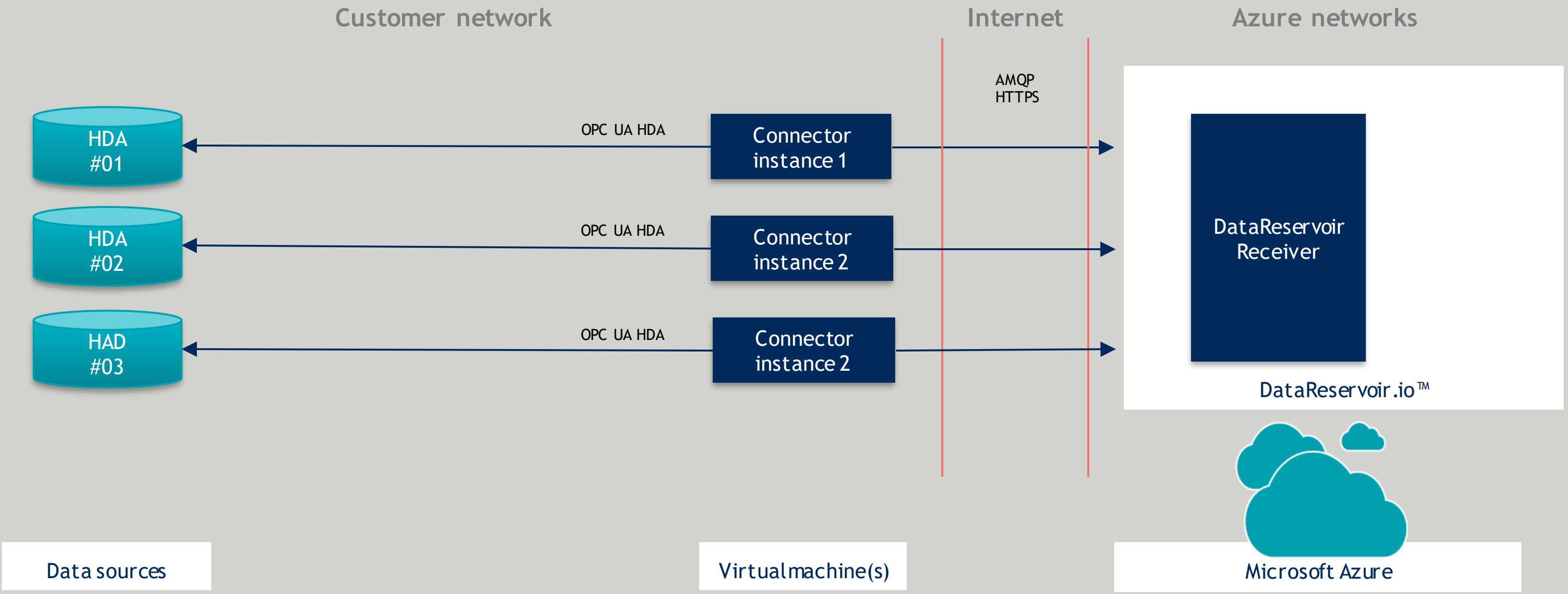
Advanced Topologies

DataReservoir.io™ Connector’s primary mode of operation is connecting to history databases over the OPC HDA protocol (classic or UA).

In cases where history databases are not available, a historian middleware should be introduced to ensure that all data can be transferred to DataReservoir in a reliable manner.

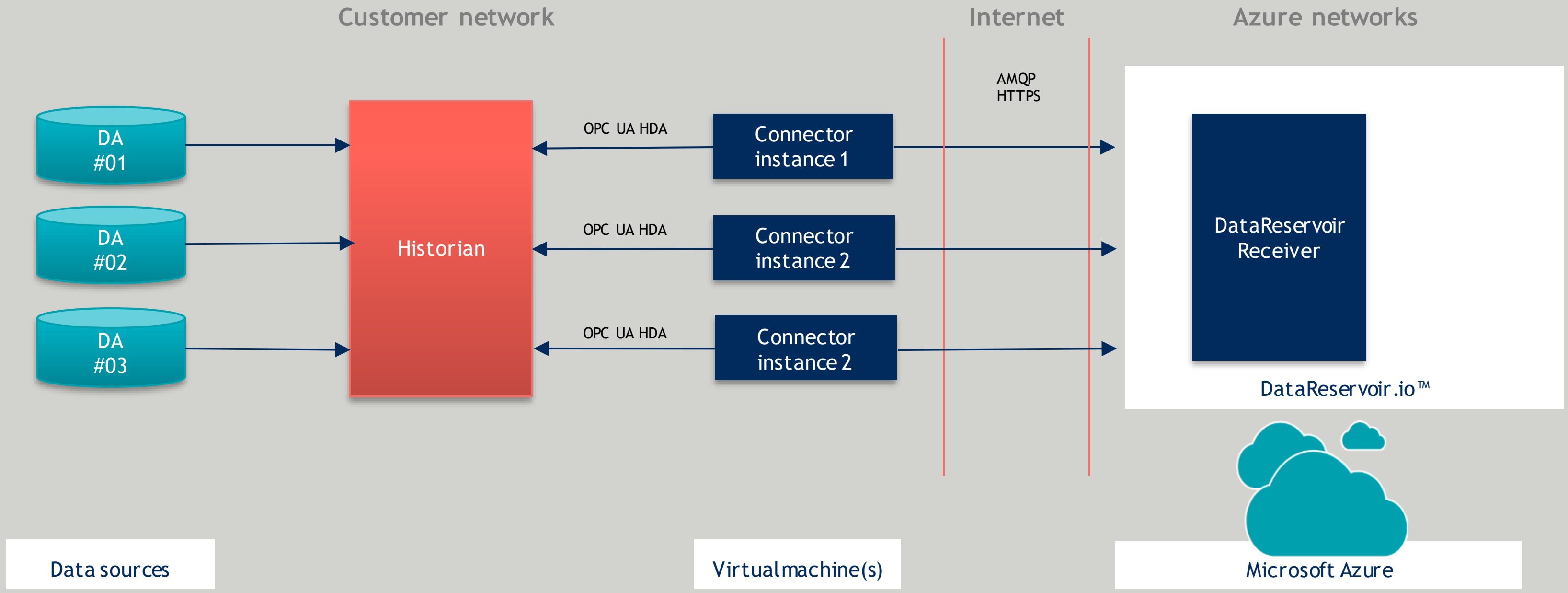
The following examples illustrate some topologies with and without history-enabled data sources.

Data Sources with HDA Support

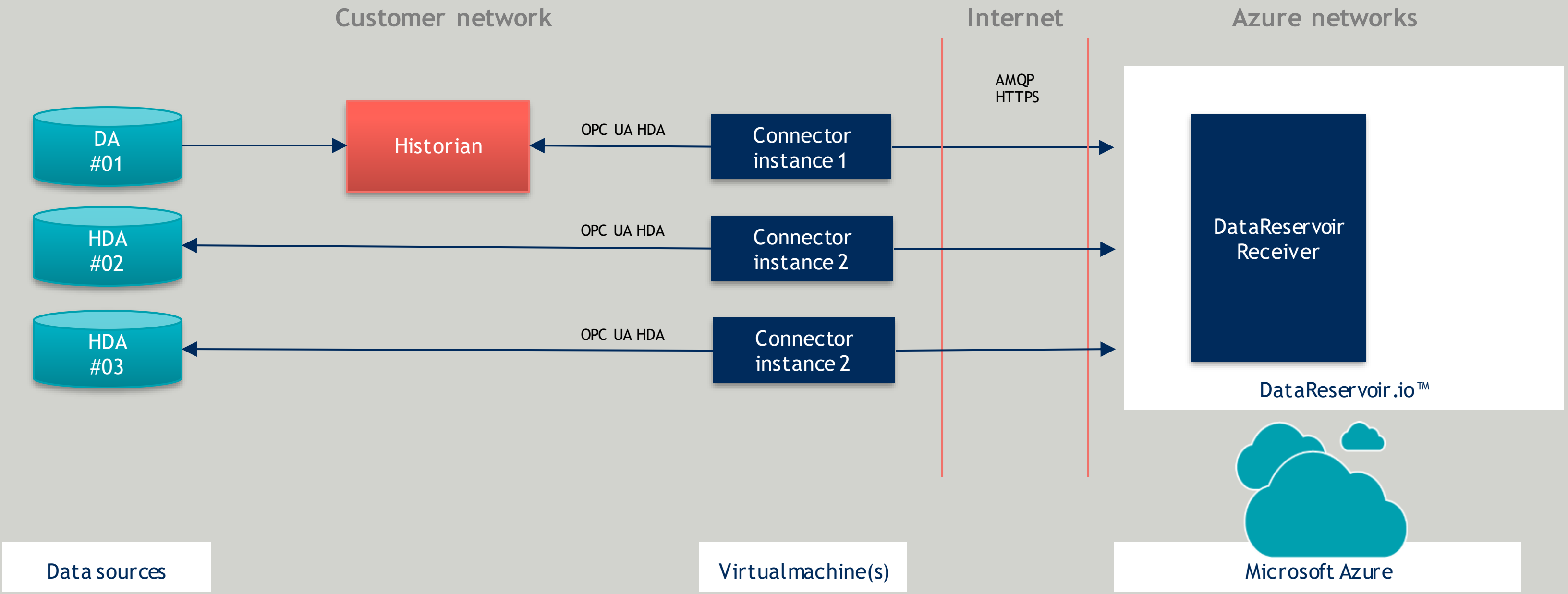


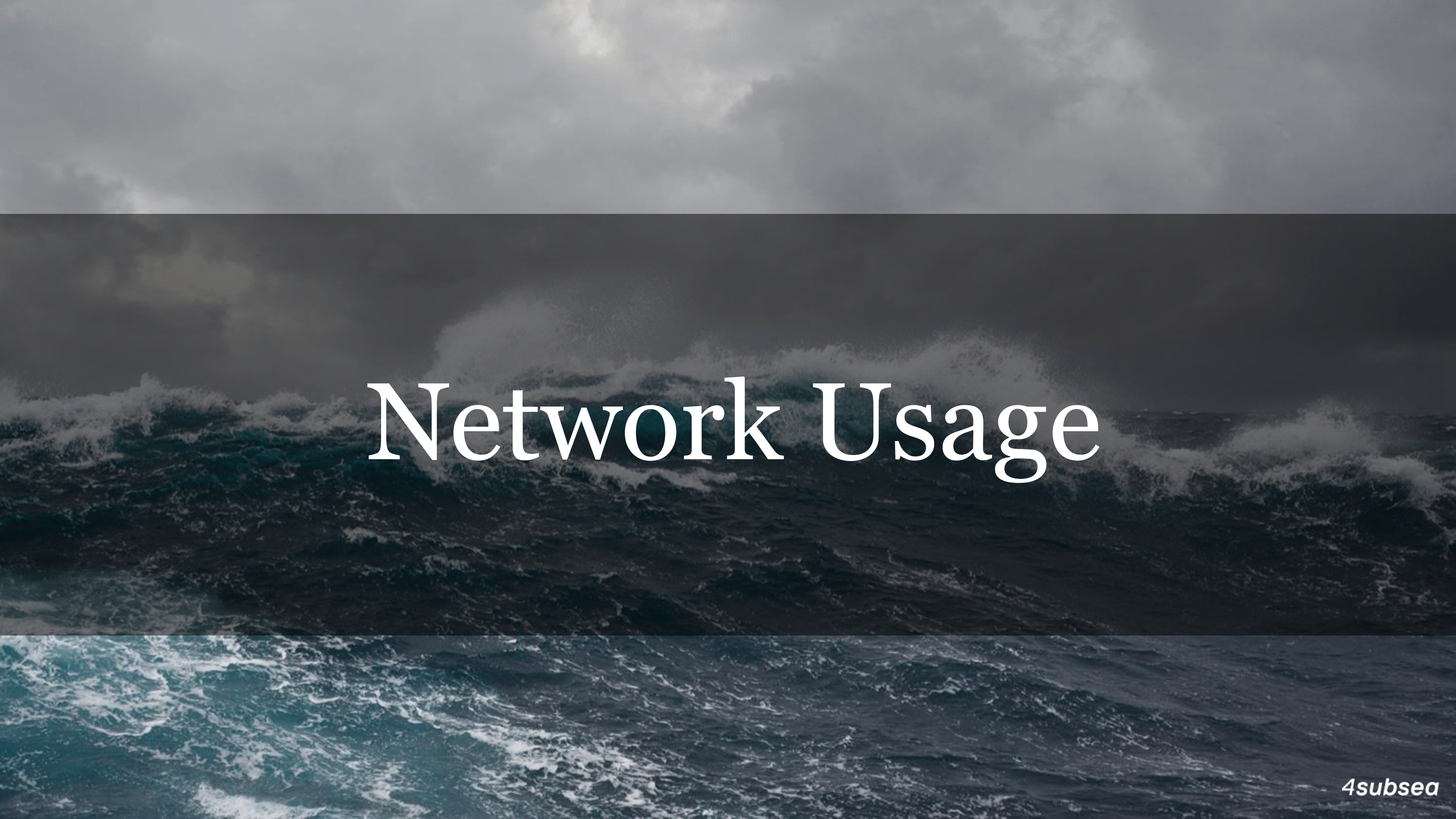
Data Sources without HDA Support

- Historian Middleware



Data Sources w. Mixed HDA Support





Network Usage

Network Usage

Network and bandwidth consumed by the Connector is highly dependent on the amount of timeseries that is configured for transfer, and their individual data frequency.

DataReservoir.io™ Connector uses two strategies to minimize network usage:

- Compression
- Value filtering

Compression

Combined with efficient use of AMPQ package transfer, all timeseries data is compressed using GZip before transmission. The achieved compression ratio will vary depending on the uniformity of the data.

Compression is enabled by default across all timeseries.

Value filtering

Value filtering enable data reduction before it is sent to DataReservoir.io™. In scenarios where timeseries have repeating or slowly changing values, DataReservoir.io™ Connector can be configured to only transmit values when they change compared to previous values.

Two modes can be configured:

- Equality: a new value that is different from the previously observed value, is transmitted. The new value is considered the new observed value.
- Threshold: a new value that is different from the previously observed value, within a certain threshold, is transmitted. The new value is considered the new observed value.

Contact 4Subsea: support@4subsea.com