Condence
Data replication framework

Technical description

7th of January 2023

Revision 6

1 Technical description

This document gives generic information about the Condence data replication framework. This document explains the principal information on the framework. The use of the document is not to provide a complete understanding of the functionality yet a basis for further discussions.

2 Introduction and general notes

Who is this product/feature for?
This product is for companies willing to automatically transfer metrics data from Condence to their data warehouse according to the required company data strategy.

Which problem are you trying to solve?
This product reduces the need for manual actions like exporting data from the Condence web UI.

What will this feature do, and what will it not do?
Automatic replication is partially manual setup by Distence. In this context, manual setup means that the monitored device and its measured metrics are selected manually to be replicated to some specified endpoint at the customer´s premises. Distence´s customers can´t do the setup at this maturity level.

2.1 Licenses

Data replication is an additional paid feature—the overall cost of using the feature cumulates from the project setup costs and monthly service fees. Refer to section 6, Deliverables.

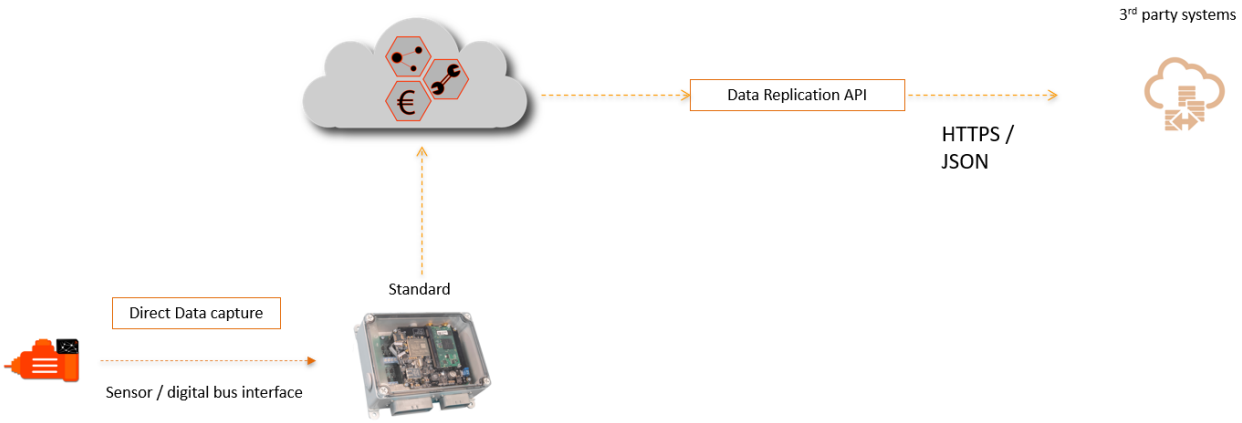
Some data, e.q. Time waveform, and wireless sensors, might require additional licences for them to be activated. For example, Condence Easy needs an additional licence for the time waveform.

3 Solution description

This chapter describes the solution framework needed for transferring data from Condence to other storage systems.

History data extraction and transfer must be configured and extracted manually, as professional work is charged separately. History data here is defined as data that exists in the Condence before automatic data transfer between systems is started.

3.1 High-level architecture



The high-level idea is to trigger data transfer to the customer´s own data storage system at the same rate metrics are received into the Condence cloud system from the smart terminals in the field. This process is called here as data replication.

The customer must have a data storage system available that the Condence data replication process can use.

The data replication process is divided roughly into four separate stages:

1. The data transfer process is triggered for defined assets/metrics when the Condence cloud receives metrics from smart terminals.
2. Outgoing metrics are converted from the Condence internal representation to the outgoing message format.
3. The message is put into the customer-specific outgoing queue.
4. Messages in the queue are sent to the customer´s data storage system when available.

3.2 Data replication triggers

Distence can configure data replication triggers manually to:

- time-series data including
  - set of data identifiers, e.g., customer, device, sensor/metrics
  - timestamp
  - value(s) measured and/or analysed at the timestamp
- notification/alarm data, including
  - set of data identifiers, e.g., customer, device, sensor/metrics
  - timestamp
  - new state identifier, e.g., “normal,” “warning,” and “damage.”
  - threshold value which triggered the state change
- raw vibration time-domain data
  - set of data identifiers, e.g., customer, device, sensor/metrics
  - timestamp
  - information about the data to be used for post-processing
  - value(s) as an array

3.3 Message conversion

Condence internal message format is not usable for other systems; therefore, message conversion is needed. Condence internal messages contain unnecessary data fields for other systems, and message conversion simplifies document representation for other systems.

3.4 Message queue

Distence creates at least one message queue per other data storage system to ensure reliable data replication. The queue will temporarily store outgoing replication messages if other data storage system receivers are down for some reason. Data is stored in the queue for a maximum of 72 hours.

3.5 Good to know

Changes in sensor configurations might affect data replication if data replication is configured to pick specific names or identifications.

4 Data transfer

Data is transmitted over an encrypted HTTPS connection. Data is transmitted in a JSON format, as specified in chapter *6. Message formats*.

5 Deliverables

1. Setup project to enable data transfer
2. The data transfer framework transfers data automatically to a specified target system in a specified format.

6 Message formats

This chapter describes the standard message formats to be delivered to the recipient system.

6.1 Time series data format

```
{
  “eventId”: “...”,
  “time”: nn.n,
  “id”: {
    “domain”: “<domain-key>”,
    “device”: “<device-key>”,
    “conf”: “<measurement-key>”,
    “target”: “<target-key>”,
    “metric”: “<metric-key>”
  },
  “path”: [ ... ],
  “groups”: { ... },
  "tags": { "tag1": "value1", "tag2": "value2" },
  “value”: nnn,
  “aggregates”: {“min”: nnn, “avg”: nnn, “max”: nnn },
  "measurement": { "quality": "<quality>", "unit": "<unit>" }
}
```

6.2 Notification/alarm data format

```
{
  “eventId”: “...”,
  “time”: nn.n,
  “id”: {
    “domain”: “<domain-key>”,
    “device”: “<device-key>”,
    “conf”: “<measurement-key>”,
    “target”: “<target-key>”,
    “metric”: “<metric-key>”
  },
  “path”: [ ... ],
  “groups”: { ... },
  "tags": { "tag1": "value1", "tag2": "value2", ... },
  "alarmName": "<name>",
  "alarmDescription": "<description>",
  “severity”: nnn,
  “severityName”: “<severity>”,
  "alarmType": "<alarm-type>",
  "alarmName": "<alarm-name>",
  “action”: “<action>”,
  “threshold”: nn.n,
  "measurement": { "quality": "<quality>", "unit": "<unit>" },
  “message”: {
    “emailSubject”: “<email-subject>”,
    “emailBodyText”: “<email-body>”,
    “emailBodyHtml”: “<email-html>”,

  }
}
```

6.3 Raw vibration time domain data format

```
{
  “eventId”: “…”,
  “time”: nn.n,
  “id”: {
    “domain”: “<domain-key>”,
    “device”: “<device-key>”,
    “conf”: “<measurement-key>”,
    “target”: “<target-key>”,
    “metric”: “<metric-key>”
  },
  “path”: [ ... ],
  “groups”: { ... },
  "tags": { "tag1": "value1", "tag2": "value2", ... },
  “report”: {
    “encoderRpm”: 1480,
    “onDemand”: true/false,
    “refId”: “...”,
    “actualSampleCount”: 65536,
    “actualSamplingFrequency”: 48828.125,
    “mVScaling”: {
      “adcMin”: -8388608,
      “adcMax”: 8388607,
      “min”: -10545.64,
      “max”: 10545.645
    },
    “settings”: {
      "channel": "IEPE_INPUT_0",
      “validRPM”: {
        “min”: 0,
        “max”: 0,
        “delta”: 0
      },
      “interval”: 3600000,
      “speedRatio”: 1
    },
    “channelSetting”: {
      "channel": "IEPE_INPUT_0",
      "sensorConversion": 1,
      “sensorQuality”: “acceleration”
    },
    “sampleCount”: 65536,
    “samplingFrequency”: 48828.125,
    “encoderRPMSource”: {
```

```
      “name”:” “,
      “fixed”: 1480
    }
  },
  “values”: […]
}
```