**Time Series**

* Time series data is a sequence of collecting data points at particular time intervals over a continuous period of time.

Eg: Sensor Data

* A Time Series data store requires a measurement with a corresponding timestamp.
* The Time Series service consists of field called an attribute field. In this field it include some additional relevant details about that specific data point, such as units or site.

Eg: “Site”:” SanFrancisco".

* Time series data consist of both regular and irregular data.
* Here some benefits on Time series:
* Efficient storage on times series data.
* Quick retrieve of data from indexing.
* Highly availability so you can access and query the data anywhere using HTTP.
* It will store Millisecond data point precision
* Horizontal scalability.

**Data Ingestion:**

This layer can provide the ability to ingest live streaming data.

**Data Query:**

This will allows to query the data using the group of data points like tags, time value as well as aggregation. We can also filter the attributes to narrow our results.

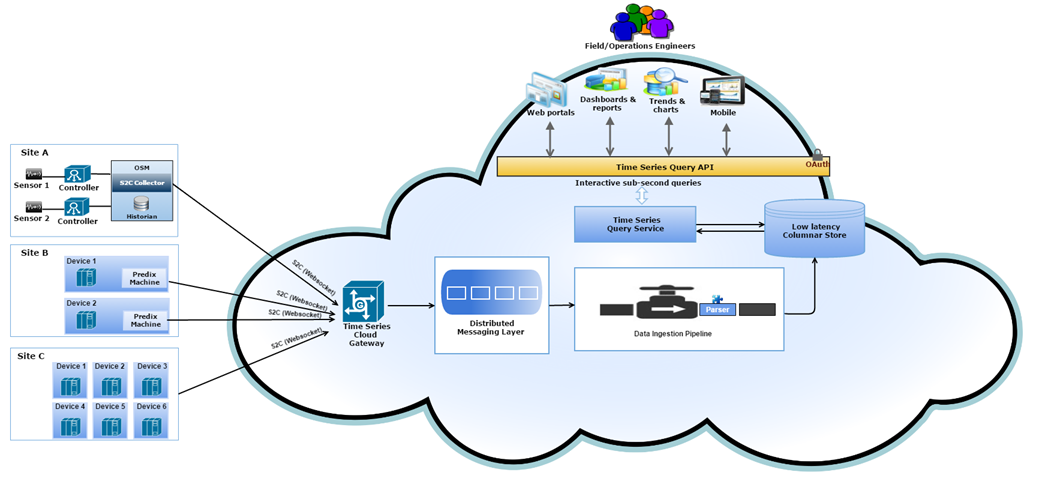


Fig: Time Series Service Architecture

* **Site A:** In this the data produce by sensor is associated with tag name and sent to the Historian for storage and management. S2C subscribes to tags and collects generated data from those tags only. S2C requests a web socket connection from a gateway application that is used for data ingestion.
* **Site B:** In this the Web Socket River establishes a connection for first attempt a data transfer and keeps that channel open for a long time . Each data transfer verifies that whether the web socket connection is open or closed. If the connection had been closed, the service opens a new connection. To communicate with the Time Series gateway, the data must be structured as shown in Pushing Time Series Data.
* **Site C:** Devices use an application to communicate directly with the web socket.

**Time Series Data Ingestion and Consumption:**

A time series uses tags, which are often used to represent sensors (for example, a temperature sensor).

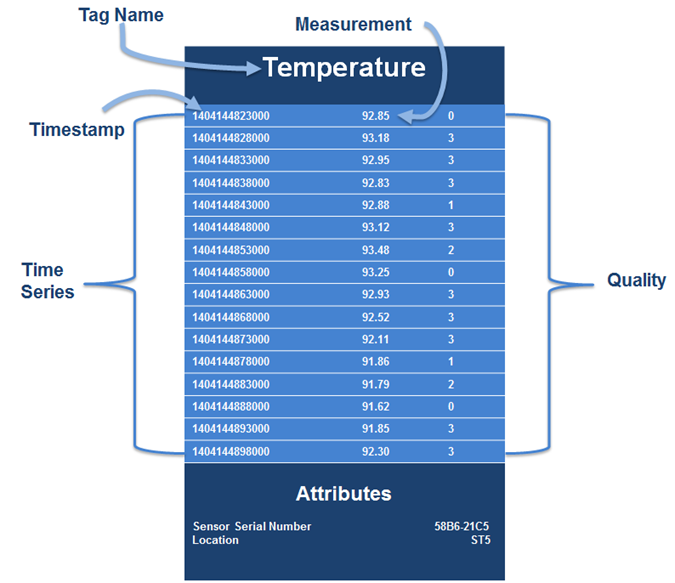


Fig: Time Series Tag Format

**Tag Name:** Tag name consists of unique name and it is required field. It may consists of some alpha numeric characters. The maximum size of the tag name is 256 characters

**Measurement:** Reading the value i.e., Number. It is required field.

**Time Stamp:** It consist of date and time with millisecond precision. It is required field.

**Quality:** It represents the quality of the data. It is an optional field. Value starts from 0 to 3

i.e., 0- Bad Quality, 1- Uncertain quality, 2- Not applicable, 3- Good quality (default value)

**Attribute:** It consists of key/ value pair that is used to store the data that is associated with tag name. It is useful to filter the data.

**Data Consumption:**

* Time series provides the REST API for aggregating and querying the time series data.
* Query will get the latest value from time series database
* It will filter by attribute values
* It will retrieve the all tags and attribute keys
* Query time series data specifying tags (sensors, etc.) and a time window
* Add aggregates and interpolate data points in a given time window.
* Delete all tags and all associated data points.

**Get Requests:**

<timeseries-app-url>/v1/aggregations //get all available aggregations

**1.1<timeseries-app-url>/v1/aggregations?fields=name="max",type="Maximum"**

//"Returns the maximum value data point for the time range"



**1.2<timeseries-app-url>/v1/aggregations?fields=name="trendmode",type="Trend Mode"**

//"Returns the maximum and minimum value data point for the time range"



1.3**<timeseries-app-url>/v1/aggregations?fields=name="count",type="Count"**

**//"Returns the number of data points"**



1.4**<timeseries-app-url>/v1/aggregations?fields=name="scale",type="Scale"**

//Scales each data point by a factor"



**1.5<timeseries-app-url>/v1/aggregations?fields=name="interpolate",type="Interpolate"**

//"Does linear interpolation for the chosen window"

****

1.6**<timeseries-app-url>/v1/aggregations?fields=name="sum",type="Sum"**

//"Adds the data points together"



**1.7<timeseries-app-url>/v1/aggregations?fields=name="diff",type="Difference"**

**//"Returns the difference between successive data points"**

****

1.8<timeseries-app-url>/v1/aggregations?fields=name="sampler",type="Sampler"

//"Returns the sampling rate of change for the data points"

**1.9<timeseries-app-url>/v1/aggregations?fields=name="div",type="Divide"**

**//"Divides each data point by a divisor"**

****

1.10<timeseries-app-url>/v1/aggregations?fields=name="min",type="Minimum"

// "Returns the minimum value data point for the time range"



**1.11<timeseries-app-url>/v1/aggregations? fields=name="avg",type="Average"**

**//Returns the average of the data point set"**

1.12<timeseries-app-url>/v1/aggregations?fields=name="least\_squares",type="Least Squares" //"Returns a best fit line through the data points using the least squares algorithm"

1.13<timeseries-app-url>/v1/aggregations?fields=name="percentile",type="Percentile"

// "Returns the percentile of the data range"

1.14<timeseries-app-url>/v1/aggregations?fields=name="dev",type="Standard Deviation"

//"Returns the standard deviation of the time series"

1.15<timeseries-app-url>/v1/aggregations?fields=name="rate",type="Rate"

//"Returns the rate of change for the data points"

**1.16<timeseries-app-url>/v1/aggregations?fields=name="gaps",type="Gaps"**

//"Marks gaps in data according to sampling rate with a null data point"



2.<timeseries-app-url>/v1/datapoints // query datapoints

**3.<timeseries-app-url>/v1/datapoints/latest** // query for current value



4.<timeseries-app-url>/v1/tags //get all tags

**Post Request**

**1.<timeseries-app-url>/v1/datapoints**

//query datapoints



2.<timeseries-app-url>/v1/datapoints/latest // query for current value