GUIDE FORM SPECIFICATION – DATA HISTORIAN

PART 1 - OVERVIEW

1.1 PURPOSE

A. The purpose of this document is to provide the necessary description of the Operational Data Historian that is a major component of the plant information management solution. This description will be used to qualify available products in terms of functionality, capability and other operating requirements. Ultimately this description will be sufficient to allow bidders to identify technology that match or exceed the desired capabilities.

1.2 SUMMARY

A. This Section describes an Operational Data Historian used for high-speed data collection and time-series data optimization. The Historian shall store process, alarm and event data for use by facility and management personnel for visualization, analytics and reporting. Historian shall be an integral component of the overall software-licensing model.

1.3 DEFINITIONS

A. API: Application Programmer Interface

B. OPC: Object Linking and Embedding for Process Control

C. OPC HDA: Object Linking and Embedding for Process Control with Historical Data Access

D. OLE DB: Object Linking and Embedding for Data Base

E. ODBC: Open Data Base Connectivity

F. Tag: A naming convention for any point within the system which includes its attributes and properties

G. UTC: Universal Time Coordination

H. Input Scaling: Convert a given measurement range to a different unit or gradient

I. Dead Banding: A neutral zone of a signal range where no action occurs

J. Rate of Change: The speed at which a variable changes over a specific period of time

1.4 SYSTEM PERFORMANCE

A. Comply with the following performance requirements:

1. The Data Historian shall provide support for a multi-node environment
2. The Data Historian shall provide a maximum write performance rate of 60 million samples/minute/node for up to 2 million tags.
3. The Data Historian shall provide a maximum read performance rate of 600 million samples/minute for up to 2 million tags.
4. The Data Historian shall provide a sustained event rate of up to 18 million/minute.
5. The Data Historian shall support up to 20,000,000 tags on a single server.
6. The Data Historian shall support up to 100,000,000 tags in a multi-server system.
7. The Data Historian shall support up to 2000 connections per server
8. The Data Historian shall be developed as a high-speed, time-based process Data Historian that does not require the use of 3rd party relational databases for operation or installation when logging time-series process data.
9. The Data Historian shall provide 1 microsecond time stamp resolution.
10. The Data Historian shall be able to store 32-bit floating point numbers, including time stamps and quality in no more than 6 bytes. If storing millisecond or OPC Quality data, then the system may occupy up to 2 additional bytes for a maximum of approximately 8 bytes per record.
11. The Data Historian shall provide automatic time-based indexing between different tags without adversely impacting storage performance. The Data Historian shall store each sample consuming only 6 bytes per sample.
12. The Data Historian shall be built with 64-bit code and run on 64-bit systems.
13. The Data Historian shall run on both Microsoft Windows based servers and Linux based systems.
14. The Data Historian shall have a method for storing in memory when the last sample was collected for each tag, assuring optimal retrieval methods for stale tags, or tags that aren't updated regularly in the system.

1.5 SERVER SYSTEM REQUIREMENTS

A. Operating System Requirements. The Data Historian shall be supported on any of the following operating systems:

1. Microsoft® Windows® Server 2022 (64-bit)
2. Microsoft® Windows® Server 2019 (64-bit)
3. Microsoft® Windows® Server 2016 (64-bit)
4. Microsoft® Windows® 11 (32-bit or 64-bit)
5. Ubuntu distributions of Linux with Docker support

B. Hardware Requirements. The system shall be capable of operating on a minimum configuration of:

1. Physical hardware or equivalent ESXi or Azure virtual machine
2. 2.8 GHz Intel Core Family Processor,
3. 8 GB RAM
4. 100 GB free hard drive space for the data archives, message files, buffer files, and log files used by the system
5. 100 Mbps TCP/IP compatible network interface adapter

1.6 SUBMITTALS

A. Product Data: Include manufacturer's technical literature for the Data Historian. Indicate:

1. Capacities
2. Performance characteristics
3. Installation instructions
4. Configuration instructions
5. Startup instructions

B. Data Communications Sources: Describe the source, interface and protocol of any data connections to the Data Historian.

C. Software Support Services: Provide support documentation on the Data Historian Manufacturer's Software Support Services, documentation shall include:

1. First year and subsequent years pricing information (basis for pricing)
2. Describe how 24/7/365 telephone support is provided
3. Describe how software revisions are provided
4. Describe how software improvements are provided

D. Training. Both on-site and off-site training may be required for personnel. The manufacturer of the equipment/software or an authorized training partner of the supplier must provide trainers and training material.

E. Installation Services. The vendor shall have the ability to provide qualified consulting and installation services. These services need to be provided either by the vendor or via certified partners. Services include:

1. Site Assessment
2. Consultation
3. Implementation
4. Assistance and turnkey
5. Validation services

F. Global Services

1. Vendor needs to be able to provide service and support the installations locally worldwide.

G. Qualification Data: Provide Data Historian manufacturer's ISO 9001-2000 certification

H. Operation and Maintenance Data: Provide Data Historian manufacturer's standard product manuals and CDs.

1.7 QUALITY ASSURANCE

A. Data Historian manufacturer shall be able to demonstrate that it has established factory test procedures.

B. Data Historian manufacturer shall be certified under ISO 9001-2008 guidelines.

1.8 DELIVERY, STORAGE, AND HANDLING

A. Data Historian software shall be delivered via electronic image download.

B. Data Historian features shall be controlled via a license management system.

C. Data Historian software shall be delivered as the latest revision at the completion of the project.

PART 2 - PRODUCTS

2.1 MANUFACTURERS

A. Manufacturers: Subject to compliance with requirements, provide products by one of the manufacturers specified.

2.2 DATA HISTORIAN

A. Manufacturers:

1. Licensing will be provided to support server operations and client access from multiple simultaneous users and will include an annualized contract for technical support and product upgrades.

2.3 ADMINISTRATION, CONFIGURATION, AND DIAGNOSTICS

A. Administrative functions (e.g., tag configuration, archive maintenance, etc.) shall be configurable via a 100% browser based interface without the need for any 3rd party ActiveX controls on the client computer.

B. A non web-based administration tool shall also be available so that a web server is not required for system configuration.

C. The user shall be able to browse and add tags from data sources (e.g. OPC, other Historians)

D. All configuration changes shall be “on-line” without the need to stop and re-start the data historian

E. System shall provide for the automatic creation of archive files and the ability to automatically overwrite the oldest archive for unattended operation

F. The system shall provide a method for backing up all on-line/active archives on-line without the need to stop the archive system.

G. The system shall provide a diagnostic dashboard function that displays the health of the system. This shall include information about data nodes, collectors, and clients.

H. All changes should be captured in human readable log files (see AUDIT TRAIL)

2.4 SECURITY

A. The system shall support expiring secure token authentication

B. The system shall support secure LDAP/Active Directory service providing single-logon capability and a central repository for information for your entire infrastructure, vastly simplifying user and computer management and providing superior access to networked resources.

C. LDAP integration shall include support for complex AD structures and very large (>32,000) AD data sets.

D. Web based client authentication shall be based on UAA, with integration with a customer's LDAP service.

E. Role-based security shall restrict user access to different administration and system functions. At a minimum, these shall include:

1. Security administrators
2. Tag maintenance
3. Archive file maintenance
4. Data collector maintenance
5. Data readers
6. Audited writes
7. Unaudited writes
8. Tag-based security shall be configurable on a tag-by-tag basis and shall include the following:

    a. Readers
    b. Writers
    c. Administrators
    d. Security system shall support both local and domain-based security
    e. System shall provide an "electronic signature" feature to validate the identity of any user for all configuration changes. All client tools provided by the vendor that have the ability to modify the system shall utilize the electronic signature feature
    f. The electronic signature system shall provide a user configurable dialog message so that the client can customize the message

2.5 AUDIT TRAIL

A. The Data Historian offer an automatic audit trail mechanism that stores all configuration changes, user connections, security violations and performance metrics.

B. The audit trail shall be stored with the data in the archive file covering the same time period so that the user only needs to manage a single file for all data and audit messages for any given period.

C. The audit trail shall not be modifiable – a user may insert custom messages, but once stored an audit message cannot be modified or deleted regardless of the user's security privileges.

2.6 SERVER COMPONENTS

A. The Data Historian shall be equipped with a System API, a client/server programming interface that maintains connectivity to the Historian Server to provide functions for data storage and retrieval in a distributed network environment.

B. The Data Historian shall be equipped with a REST API, allowing third party client access to all Historian functions, including tag creation and configuration, and all query methods to expose collected data. The REST API shall support both querying of data and writing of data values.

C. The Data Historian shall be equipped with a Java API, allowing third party client access to all Historian functions, including tag creation and configuration, and all query methods to expose collected data

D. The Data Historian shall be equipped with a .NET Client Access API. The Client Access API supports both 32-bit and 64-bit Windows Operating Systems.

E. The Data Historian shall be equipped with a catalog of data collectors that have the ability to connect to multiple sources of data and send data either to the local Historian or to a cloud-based time series system. Data sources should include OPC UA, OPC DA, OPC HDA, ODBC, MQTT, other Historians such as Aveva Historian and OSIsoft's PI, and a file collector.

F. The Data Historian shall be equipped with a Collector Toolkit, which allows rapid development of new collectors. The Collector Toolkit shall support the development of both Windows-based and Linux-based Collectors.

G. The Data Historian shall be deployable across multiple independent mirrored nodes to run in a highly available and load-balanced environment.

H. The Data Historian shall include a Model Service which enables users to create a model context mapping for the tags stored in the Historian. Applications may read data in the context of the model via REST o or OPC UA interfaces.

I. The Data Historian shall include an OPC HDA Server to read the raw data stored in Historian and send it to the interested OPC HDA clients. OPC HDA compliant clients can connect to the Historian HDA Server to acquire and analyze data.

J. The Data Historian shall include an OPC UA Server to read the raw data stored in Historian and send it to the interested OPC UA HDA or OPC UA DA clients. OPC UA compliant clients can connect to the Historian UA Server to acquire and analyze data.

K. The Data Historian shall provide customers with the ability to store, organize, and manage tags according to the data source and storage requirements. This function shall support both logical and physical storage.

2.7 TAG AND DATA COLLECTION

A. The system shall provide a graphical interface to browse and add tags from any supported data source. Added tags shall automatically determine the data type, description, tag name from the data source

B. The system shall provide APIs which may be used to create and maintain tag definitions. APIs shall include .Net, C++, REST, and Java.

C. All configuration changes shall be performed on-line with no restart required.

D. The system shall utilize UTC for time stamping data so that it is unaffected by daylight savings time or regional time differences.

E. Data collectors shall support synchronizing time stamps to a central server time so that all data has the same relative time stamps.

F. The Data Historian shall be able to reject storing data where the system determines time synchronization failure.

G. The Data Historian shall allow storing of data with future timestamps (predictions).

H. System shall support both polled and exception-based data collection.

I. The minimum polled collection interval shall be 100 milliseconds.

J. The minimum for exception-based collection shall be 1 millisecond.

K. Unsolicited tags shall have a method to throttle/limit the rate of unsolicited data writes. E.g., no faster than every 5 seconds.

L. System shall provide both a collection rate and a collection-offset configuration.

M. Collection rates shall be configurable using intuitive seconds, minutes, hours drop down select lists.

N. System shall support 1 microsecond time stamp resolution.

O. System shall support the following native data types:

1. Single integer (2 bytes)
2. Double integer (4 bytes)
3. Quad integer (8 bytes)
4. Unsigned single integer (2 bytes)
5. Unsigned double integer (4 bytes)
6. Unsigned quad integer (8 bytes)Byte (1 byte)
7. Single Float (4 bytes)
8. Double Float (8 bytes)
9. Boolean (1 byte)
10. Scaled floats (scaling a float across 2 bytes)
11. Fixed length string (of any length)
12. Variable length string (of any length)
13. Binary Large Objects (BLOBs of any size)

P. System shall provide for input scaling (e.g. automatically scale a 0-4096 input to 32-212 degrees F)

Q. System shall provide dead banding compression algorithm (+/- limits around the process value)

R. The dead banding compression system shall be performed on the remote data collection PCs so that values that do not exceed the dead band are not reported to the server and do not consume network bandwidth.

S. The dead banding shall have a “time out” feature so that a value is stored at a regular interval regardless if it has not exceeded the dead band for polled tags.

T. The dead banding shall be configurable as a % of the engineering limits.

U. System shall provide a rate of change compression algorithm (+/- limits around the real-time slope of the process)

V. The rate of change algorithm shall have a “time out” feature so that a value is stored at a regular interval regardless if the slope deviation has not been exceeded.

W. The rate of change compression limits shall be configurable as a % of the engineering limits.

X. System shall ensure that data spikes are properly represented in trend and client tools. E.g., if a value has been stead state for several periods and suddenly spikes to a high value trends shall properly display this as a spike/step change and not a ramp.

Y. System shall provide for recording time stamps from either a collector PC or from the OPC/PLC tag.

Z. Data collectors shall have an automatic store and forward mechanism to ensure that data is not lost when disconnected to the data archiver.

AA. The store and forward mechanism shall not require the user to pre-allocate a buffer file or set a maximum buffer file size. Instead, the system shall provide for utilizing available disk space up to a user configurable limit.

BB. The store and forward mechanism shall automatically detect when the archiver is available and forward data from the buffer files while simultaneously collecting all incoming data.

CC. Data collectors shall not require a hardware key or other licensing on the collection PC.

DD. Once installed, data collectors shall automatically register and configure themselves with the data archiver service without the need for any additional configuration on the archiver.

EE. Management of a collector installed on a remote machine shall be able to be performed from the Data Historian server administrator utility. Management includes:

1. Starting the service
2. Stopping the service
3. Starting the data collection
4. Pausing the data collection
5. Changing configuration
6. Configuring additional instances of installed collectors
7. Deleting or moving data buffer files

FF. A user shall be able to store comments with any collected data.

2.8 PRE-PROCESSING AT THE DATA COLLECTOR

A. The system shall have a pre-processing engine that allows the user to store the results of a derived value on one or more sources, without storing the raw source itself.

B. The pre-processor shall work on any collector (except the file collector)

C. The pre-processor shall allow for basic math functions on one or more tags.

D. The pre-processor shall provide a basic functions library.

E. The pre-processor shall provide a unit of measure conversion library.

F. The pre-processor shall provide a browser-based UI that allows the user to create the derived value.

2.9 CALCULATION ENGINE

A. Architecture and Data Collection

1. The system shall provide a calculation engine for the automatic calculation and analysis of both incoming and archived data and then storing the results of the calculation in the data historian as a tag value.
2. The calculation engine shall be configurable so that it can be installed on a PC, or several PCs that are remote from the data historian.
3. The calculation engine shall have a store and forward mechanism to ensure that calculated results are not lost if the connection to the data historian is down.
4. The calculation engine shall have a recovery logic system so that any updates to trigger tags (calculation inputs) cause the calculation to re-fire. The calculation tags shall be configurable to execute both as polled or unsolicited/trigger-based tags.
5. The minimum poll rate shall be 100 milliseconds.
6. Unsolicited tags execute their calculation whenever an assigned trigger tag receives a new value, time stamp or change in quality.
7. Calculations shall support an unlimited number of trigger tags.

B. Calculation Functions

1. The system shall support the following calculation functions:

    a. Browse and select any historian tag as an input to the calculation
    b. Current/last stored value of any tag
    c. Previous values of a tag
    d. Next value of a tag
    e. Interpolated values of a tag
    f. Current/previous/next time stamp
    g. Current/previous/next quality
    h. Historical time-weighted average
    i. Historical time-weighted minimum
    j. Historical maximum

k. Historical time-weighted standard deviation
l. Historical time-weighted total
m. Historical count of samples
n. Historical raw average
o. Historical raw standard deviation
p. Historical raw total
q. Time of minimum sample
r. Time of maximum sample
s. Total time that a sample was good
t. The system shall support filtered calculations so that and of the above calculations are automatically filtered/limited based upon another tag's value. E.g., Calculate the minimum amps (tag 1) over the previous day, but exclude any samples in which for the same time period the line voltage (tag 2) was 0. Or, return the average for tag 1, while the batch id (tag 2) = 'ABC'

2.10 CALCULATION CONFIGURATIONS

A. The system shall support filtered calculations so that any of the above calculations are automatically filtered /limited based upon another tag's value. E.g., Calculate the minimum amps (tag 1) over the previous day, but exclude any samples in which for the same time period the line voltage (tag 2) was 0. Or, return the average for tag 1, while the batch id (tag 2) = 'ABC'

B. Calculations shall support full visual basic scripting within the calculation.

C. Calculations shall support Python expressions.

D. Calculations shall be configurable using the same administration tools (web and non-web) including all visual basic scripting and functions.

E. The system shall have tools to assess the time that a calculation takes to execute, as well as a means of disabling calculations that exceed a configurable maximum execution time.

F. The calculation shall be stored in the audit trail along with a full history of modifications and the time, date and user that made the modifications.

G. The calculation engine shall have a manual recalculation engine so that calculations can be applied to legacy data and values stored alongside the legacy data as if the calculation engine had been executing in real time when the legacy data was archived.

2.11 SERVER-TO-SERVER ENGINE

A. The system shall have a server-to-server engine so that tag data can be automatically forwarded from a one historian to a remote historian.

B. The server-to-server engine shall support all of the functions and features described in the section on the "Calculation Engine" (e.g., calculations, visual basic scripting, re-calculation, etc.)

C. The server-to-server engine shall provide for the forwarding of audit messages as well as data.

D. The server-to-server engine shall provide for tag browse and configuration from the destination server's administration interface. The user shall not need to configure server-to-server tags at the remote historian.

E. The server-to-server engine shall provide a mechanism for data packet compression to reduce data transmission bandwidth consumption.

2.12 Alarm and Event storage

A. The Data Historian shall provide the capability to store Alarm & Event data generated by OPC sources.

B. The Data Historian shall support data collection from multiple OPC A&E servers

C. The Data Historian shall provide query mechanism that support retrieval of Alarm details, Alarm history, and Events.

D. Alarm and Event data shall be able to be queried through the vendors Excel Add-in.

E. Alarm and Event data shale be able to be queried through the vendors OLE DB provider.

F. The Data Historian shall provide administrative functions to back up, restore, and purge Alarm and Event records from the database.

2.13 DATA EXPORT AND IMPORT TOOLS

A. The Data Historian shall provide a high-performance data export utility that includes deadband compression support.

B. The export utility shall support the exporting of both time series data and Alarm & Event data

C. The export utility shall support time-series data export from the vendor's Data Historian along with ODBC data sources and 3rd party Data Historian products.

D. Exported data shall be stored in text files

E. The Export utility shall automatically group data files on a periodic basis and compress the grouped files for efficient transport.

F. The Data Historian shall provide a configurable FTP based mechanism, and support Background Intelligent Transfer Service for moving exported files to a destination server.

G. The Data Historian shall provide a high-performance data file import utility capable of importing a minimum of 20,000 samples per second.

H. The file import utility shall support both time-series data and Alarm & Event data

I. The Data Historian shall have a native mechanism for sending data to Microsoft Azure IoT Hub.

2.14 HIGH AVAILABLE, LOAD BALANCING, AND HDFS

A. The Data Historian shall index all information by tag and time stamp for fast data retrieval.

B. The Data Historian shall be able to concurrently write the same tag and time stamp to multiple nodes.

C. The Data Historian shall support buffering if either node is unavailable, which will work as described in the Tag and Data Collection section.

D. The Data Historian shall support "round robin" of read requests across multiple nodes to improve read performance.

E. The Data Historian shall have the native capability to utilize HDFS System for "big data" queries. The HDFS System shall have a web interface to support both MapReduce and HAWQ queries. The Historian should support Hadoop, Cloudera, and Pivotal distribution systems.

F. The HDFS System shall have an extensibility framework to support model-enabled queries, the snap-in of third-party analytic tools like MATLAB's, and the ability to transform native Historian archives into a tabular, open source format, that maintains a high level of compression.

2.15 CLIENT TOOLS

A. Microsoft Excel

1. The data historian shall come with a Microsoft Excel tool bar so that users can readily extract data and develop reports using Excel.
2. The Excel add-in shall provide for the ability to import or modify tags and other configuration data.
3. The Excel add-in shall provide for the ability to import or modify stored data.
4. The Excel add-in shall provide for the ability to import and view comments for any stored data
5. The Excel add-in shall not require the user to know the SQL query language.
6. The Excel interface shall provide for automatic recalculation if any cell is changed – e.g., a tag name, time periods, selected calculations and so forth.
7. The Excel interface shall be supplied with sample reports that can be customized by the user.

B. Web Client / Visualization and Analysis Operations Hub

1. The Data Historian shall provide an HTML5 based client that provides users with the ability to develop, manage, and deliver applications that collect, display, and analyze data from equipment or servers.
2. The Data Historian HTML5 Client shall support Data Management: Modeling including Entities and Pivot Tables; Queries, events, and notifications
3. The Data Historian HTML5 Client shall include a user-friendly WYSIWYG App Builder with a rich development library including extensive widgets and drag-and-drop configuration. Responsive layouts per page (Web, mobile, tablet). Actions, Conditions, and Formulas. QR code navigation for mobile.
4. The Data Historian HTML5 Client shall support both asset model and flat tag support
5. The Data Historian HTML5 Client shall include an Advanced Historian Trend Analysis App which provides ad hoc trending and KPI dashboards from the Data Historian data.
6. The Data Historian HTML5 Client shall allow data analysis in model context with app capabilities such as out-of-context model including pen support, favorites, annotations, multiple slider support, statistics (Min, Max, Avg, Std Dev and Total), print screen, and export to CSV
7. The Data Historian HTML5 Client shall provide task lists on an asset with context – view and act on tasks in real time with Proficy Workflow integration
8. The Data Historian HTML5 Client shall support role-based permissions at the page level
9. The Data Historian HTML5 Client security model shall include AD and shared UAA support
10. The Data Historian HTML5 Client shall allow triggering of automatic action based on events and data
11. The Data Historian HTML5 Client shall support event actions, such as sending an email (templates included), running a query, and sending a command to a device
12. The Data Historian HTML5 Client shall provide the ability to create entities and queries for a relational database
13. The Data Historian HTML5 Client shall support platform independence by using HTML5 and CSS3
14. The Data Historian HTML5 Client shall provide the ability to access applications using PC displays or mobile devices
15. The Data Historian HTML5 Client shall provide the ability to control access to an application and data, based on user roles
16. The Data Historian HTML5 Client shall support query types, such as “Get,” “Update,” “Insert,” and “Delete”
17. The Data Historian HTML5 Client shall include an Excel Add-in for detailed queries by model, asset, type, and flat tag
18. The Data Historian HTML5 Client shall provide an SDK for custom widgets and entities
19. The Data Historian HTML5 Client shall support up to 1,000 concurrent clients on a single server

20. The Data Historian HTML5 Client shall provide multi-language support: English, Japanese, Chinese, Korean, Spanish, German, Russian
21. The Data Historian HTML5 Client shall provide a library of widgets
    1. Inputs: Check box, radio button, combo
    2. Displays: Text, images, charts, graphs, grids, data tables, maps, lists, gauges, indicators, timeline chart
    3. Layouts: Separators, containers, lines
    4. Date/time picker
    5. Trend
22. Connectivity
    1. OPC UA
    2. Relational databases
    3. Time series
    4. Vendor's Data Historian
    5. MQTT: A machine-to-machine protocol using a lightweight publish/subscribe messaging transport
    6. REST API: A method of allowing communication between a Web-based client and server
    7. URL

C. OLE DB

1. The data historian shall come with an OLE DB interface so that data can be extracted and viewed by client applications such as SQL Server and Crystal Reports.
2. The OLE DB interface shall provide access to all server time-weighted calculations as described in the Microsoft Excel section above.
3. The OLE DB Provider has read-only access. You cannot insert, update, or delete data in archives using the provider.
4. The OLE DB interface shall provide SET statements and other functions so that time-weighted calculations and reports can be easily extracted.
    a. e.g. The user shall be able to get the hourly average of a tag or several tags, regardless of how frequently they were collected by entering a statement such as shown below:
    b. SET Interval milliseconds = 1Hour, Start Time = Yesterday, End Time = Today, Calculation Mode = Average Select Timestamp, Value from Raw Data where Tag name = Tag1 or Tag name = Tag2 SET statements shall include functions as shown below:
5. SET statements shall include functions as shown below:
    a. Today, Yesterday, Now, Beginning of month, etc.

b. Interval = hours, minutes, seconds (for evenly spaced reports)
c. The OLE DB interface shall provide access to all system configuration, audit messages, and data
d. The data historian shall come with an OPC HDA interface so that data can be extracted and viewed by OPC HDA client applications.

2.16 DOCUMENTATION

A. The system shall provide complete user documentation, including examples of how to operate the various modules within the system. The documentation must be in electronic format, HTML based with the ability to search for topics by keyword or search for specific text.

B. The online help system in the product needs to be context sensitive such that immediate help is available for the selected functionality.

C. Product Documentation must be available on the vendor's web site for users to download.

2.17 HMI/SCADA

A. The data historian shall have native connectivity to that manufacturers HMI/SCADA product.

B. Tags configured within the HMI/SCADA system shall be able to be automatically added to the Data Historian.

C. The HMI/SCADA system shall use data stored in the Data Historian to populate trend displays.

D. The HMI/SCADA system shall use data stored in the Data Historian to enable playback of HMI/SCADA screens.

E. The HMI interface shall provide the user with a drop-down list to specify different time zones the data shall be represented (since the historian stores data as UTC it can be represented in any relative time zone)

F. The HMI interface shall provide the ability to select if charts shall reflect time changes due to daylight savings time. E.g., when the clocks move forward in the spring the HMI chart shall display 1:00 AM and then 3:00 AM.

2.18 VENDOR REQURIEMENTS

A. Development Life Cycle

1. The vendor shall have an established and documented development life cycle procedure that allows for traceability of features and functions throughout that life cycle. Traceability of features shall be maintained from Marketing Functional Requirements through Design,

Development, Help Development and Testing. Finally, requirements traceability matrix is maintained through the development lifecycle and shall track customer requirements, development risk and corrective action.

2. The vendor shall have a formal and documented set of quality assurance procedures that are applied to the engineering design, development, and documentation of the software, including validation. The presence of a formal quality assurance department shall be required.
3. The vendor shall follow a documented configuration management system.
4. The vendor shall also demonstrate that its source code for the product is regularly archived both on-site and off-site in facilities suitable to withstand physical harm.
5. The vendor shall allow (with reasonable protection like a non-disclosure agreement) for on-site auditing of the development life cycle to ensure good practice.
6. The vendor shall have a robust Secure Development process that includes design practices, code reviews by security experts and penetration testing.

B. Product Life Cycle / Obsolescence

1. The vendor shall have an established product life cycle policy, which is published and available.
2. The vendor of the supplied product shall have a formal obsolescence policy that describes the expected product life cycle. This needs to include a declaration of a time period between product maturity and ultimately last buy opportunities.

C. Manufacturer Qualifications

1. The manufacturer shall have shown a high commitment to product, manufacturing and design process quality.
2. The manufacturer shall have fully operational quality assurance and quality control programs in place. Complete documentation describing the quality assurance and quality plan shall be available.

D. ISO 9001 Certified

1. The vendor shall be able to demonstrate that it has established procedures
2. Vendor needs to be certified under the ISO 9001 guidelines