

SoW RFP Appendix A

CVALNG Fall 2021 SCADA Framework RFP Checklist

Communication and Data

- ☐ Ability to with OPTO22 PLCs, HMI, Supervisory workstations, Operator Systems, and an open source Data Historian.
- ☐ Ability to use OPC-UA.
- ☐ Sub-framework for creating third-party connections to devices and PLCs.
 - ☐ Windows .NET
- ☐ Built-in support for connecting to common databases such as the following:
 - ☐ MySQL
 - ☐ MSFT SQL Server
 - ☐ PostgreSQL
 - ☐ Other (Requires customer acceptance)
- ☐ A sub-framework for implementing connections to other databases or kinds of databases.
- ☐ Data historian for logging time series data to a database.
- ☐ A open source distributed data historian with the ability to read and write data across multiple physical machines.
- ☐ A way to forward data being collected from a remote location to another location or central server.
- ☐ Ability to access and use SOAP Web Services provided by external applications.
- ☐ Ability to create SOAP Web Services that can be accessed and used by external applications.
- ☐ Ability to access and use HTTP-based REST APIs provided by external applications.

- ☐ Ability to create HTTP-based REST APIs that can be accessed and used by external applications.
- ☐ Full support for sending and receiving email, including attachments.
- ☐ Ways for applications built using the SCADA framework to communicate with each other.

Analyze, Evaluate and Take Action Based on Data

- ☐ A scripting or programming language for creating algorithms to analyze, evaluate and take action based on data, including preparation for follow-on integration of AI and ML technologies.
- ☐ Pre-built tools that can be configured to receive data, process it and take action.

Architecture

- ☐ When client applications are modified server software sends updates to all client applications. Client applications are easily modified and updated over a computer network.
- ☐ Functionality for separating, organizing, monitoring and managing client and server applications.
- ☐ Multiple instances of server software can communicate and share data.
- ☐ Redundancy functionality. Client applications can connect to other servers when communication connections are lost.
- ☐ Easy way for the SCADA framework to be upgraded when new versions of the software are released.
- ☐ The SCADA framework can run on the following operating systems:
 - ☐ Windows
 - ☐ OS X
 - ☐ Linux
 - ☐ Unix

- ☐ Android
- ☐ iOS
- ☐ Other
- ☐ Applications developed with the SCADA framework can run on the following operating systems:
 - ☐ Windows
 - ☐ OS X
 - ☐ Linux
 - ☐ Unix
 - ☐ Android
 - ☐ iOS
 - ☐ Other
- ☐ The SCADA framework can run on mobile devices.
- ☐ Applications developed with the SCADA framework can run on mobile devices.
- ☐ Extensibility. New tools and functionality can be created and added to the SCADA framework by third-party developers.

Creating Applications

- ☐ Software libraries for creating and testing applications.
- ☐ Development environment for creating and testing applications.
- ☐ Other tools for creating and testing applications.
- ☐ Pre-built GUI components for creating the following:
 - ☐ HMI screens
 - ☐ Information dashboards
 - ☐ Business process workflows
- ☐ Support for creating, displaying and using PDF reports.

- ☐ Support for creating, displaying and using reports in other document formats.
- ☐ Scripting or programming language for the following:
 - ☐ Creating new GUI components.
 - ☐ Custom functionality
 - ☐ Customizing components
 - ☐ Connecting data between components and different parts of an application.
- ☐ Ways to reuse functionality or code inside a single application.
- ☐ Ways to reuse functionality or code across multiple applications.
- ☐ Example applications.
- ☐ Template applications that use common strategies and/or application structure.
- ☐ Sets of images and art that can be used in applications.
- ☐ Sets of icons that can be used in applications.
- ☐ Drawing tools
- ☐ Ability to use images and drawings that have been created outside the SCADA framework.
- ☐ Ability to run and test applications before deploying or making them live.
- ☐ Debugging tools. Ways to log errors and other data that is used for debugging and understanding applications.
- ☐ Revision control
 - ☐ Ability to see past changes to an applications.
 - ☐ Ability to revert applications to earlier states of development.
- ☐ Ways to make backups of applications and important configuration information.

Alarms

- ☐ Built-in functionality and tools for creating and handling alarms.
- ☐ Tools for notifying people of alarms through the following:

- ☐ Email
- ☐ Phone calls
- ☐ SMS texts

Security

- ☐ Option for using TSL/SSL for encrypting data sent over computer networks.
- ☐ Built-in functionality for user authentication and assigning permissions to users:
 - ☐ Permissions for parts of client applications
 - ☐ Permissions for accessing data
 - ☐ Permissions for accessing parts of the server software
 - ☐ Permissions for accessing server applications
- ☐ Ability to tie into and use existing authentication and user account systems:
 - ☐ Active Directory
 - ☐ Other user account systems
- ☐ Logging user actions to a database.
- ☐ General logging of anything important that occurs.
- ☐ A way to be notified of security updates about the SCADA framework.
- ☐ Code signing technology to verify that any upgrades, new tools or functionality added to the SCADA come from a correct source and have not been tampered with or corrupted.

Licensing

- ☐ A way to purchase licenses for different parts and tools of the SCADA framework. This way people can purchase just what they need.
- ☐ An easy way to view and manage licenses and install and uninstall them.
- ☐ Licenses should be reasonably priced and arranged in such a way that as SCADA

systems grow, the licensing scales with the systems and does not become too costly to become prohibitive.

- ☐ Corporate licensing.
- ☐ Site licensing.

Support

- ☐ Good tech support options.
- ☐ Good ways to learn about the SCADA framework.
- ☐ Confidence that the SCADA framework will not be abandoned or given lower priority in the future.
- ☐ Active development by a stable technical team.
- ☐ New features and improvements are added to the SCADA framework based on what people are asking for.
- ☐ Continuous effort on the development team to find and fix bugs that are found and to generally improve the software, and help people who are using the software.

This list was created by Nick Mudge from [Perfect Abstractions](http://www.perfectabstractions.com). Feel free to modify and use this list to evaluate SCADA frameworks. More information about this list can be found here: <http://www.perfectabstractions.com/blog/a-software-framework-for-scada-applications>