NIST Advanced Manufacturing Series 300-2

Software Requirements Specification to Distribute Manufacturing Data

Thomas Hedberg, Jr.
Moneer Helu
Marcus Newrock

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AMS.300-2

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Advanced Manufacturing Series 300-2

Software Requirements Specification to Distribute Manufacturing Data

Thomas Hedberg, Jr.
Moneer Helu
Systems Integration Division
Engineering Laboratory

Marcus Newrock
Office of Data and Informatics
Material Measurement Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AMS.300-2

December 2017

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

Contents

List of Figures

1. Introduction

This section of the Software Requirements Specification (SRS) document provides the purpose of the SRS, copyright and commercial systems disclaimer, scope of the SRS, acronyms and abbreviations used in the SRS, the definition of verbal forms, list of references, and an overview of the SRS document's structure.

1.1 Purpose

The purpose of this SRS document is to describe the requirements specifications for applications and a data repository to distribute manufacturing data.

Prospective developers, technical-assessment personnel, and interested end-users are the intended audience of this SRS document.

1.2 Disclaimer

The work presented in this SRS is an official contribution of the National Institute of Standards and Technology (NIST) and not subject to copyright in the United States. Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST. Nor does it imply that the products identified are necessarily the best available for the purpose.

1.3 Scope

A software system to distribute manufacturing data via a web service is the application being defined by this document. The software system contains two component applications. The first component is an application to stream time-synchronized manufacturing data, herein referred to as the Volatile Data Stream (VDS). The second component is an application for querying a database containing manufacturing data, herein referred to as the Query-able Data Repository (QDR). The two components combined are herein referred to as the "software system."

The software system enables researchers, practitioners, and solution providers to interact with real data generated from a manufacturing facility. The VDS supports real-time data-monitoring activities. The QDR enables object-of-interest observation. The data distributed via the software system enables research in manufacturing-related domains. The distributed data also supports the optimization and enhancement of practitioner processes. Lastly, the distributed data supports solution-provider development activities, such as designing, coding, testing, and debugging software applications.

1.4 Acronyms and abbreviations

API Application Programming Interface 6, 7, 9, 11–13

ASCII American Standard Code for Information Interchange 9–11

CRUD Create, Retrieve, Update, and Delete 9, 12, 13

CSV Comma Separated Value 8, 10, 11

HTML HyperText Markup Language 7

HTTP HyperText Transfer Protocol 4

HTTPS HTTP Secure 10

JSON JavaScript Object Notation 8, 10, 11

MTBF Mean Time between Failures 14

MTTF Mean Time to Failures 14

MVC Model-View-Controller 8

NIST National Institute of Standards and Technology 1

NoSQL Not only Structured Query Language 6

PDF Portable Document Format 7

QDR Query-able Data Repository 1, 4, 6–12, 14, 15

QIF Quality Information Framework 9

REST Representational State Transfer 6, 7, 9–13

SRS Software Requirements Specification 1, 3, 4, 6, 8, 11, 12, 14

SSL Secure Sockets Layer 6

TLS Transport Layer Security 6

UI User Interface 6, 8, 9, 12

UUID Universally Unique Identifier 13

VDS Volatile Data Stream 1, 4, 6–11, 14, 15

XML Extensible Markup Language 4, 8–11

XSD XML Schema Definition 4, 8, 13

XSLT Extensible Stylesheet Lanaguage Transformation 8

1.5 Verbal Forms

1.5.1 Must

The verbal form "must" indicates requirements to be followed strictly to conform to this document and from which no deviation is permitted. Must is synonymous with "shall."

1.5.2 Should

The verbal form "should" indicates that a possibility among a set of possibilities is recommended as particularly suitable (without mentioning or excluding other possibilities) or that a certain course of action is preferred but not necessarily required.

1.5.3 May

The verbal form "may" indicates a course of action permissible within the limits of the document.

1.6 References

[1] MTConnect Institute (2014, Last Accessed June 2017) MTConnect Standard, Version 1.3, Standard. URL http://www.mtconnect.org/standard-documents.

[2] MTConnect Institute (2017, Last Accessed June 2017) Reference MTConnect Schemas, Standard. URL http://schemas.mtconnect.org/.

[3] IEEE Standards Association (2011, Last Accessed June 2017) Systems and software engineering – Life cycle processes –Requirements engineering, Standard. (IEEE 29148-2011) URL http://standards.ieee.org/findstds/standard/29148-2011.html.

[4] World Wide Web Consortium (2011, Last Accessed June 2017) REST, Web Page. URL http://www.w3.org/2001/sw/wiki/REST.

1.7 Overview

The remainder of this SRS contains an general description of the software system (Section 2) and the specific requirements for the software system (Sections 3, 4, and 5).

2. General Description

This section of the SRS describes the general requirements that drive the design of the software system. The goal is not to state specific requirements, but rather to provide context to make those requirements easier to understand.

2.1 System Perspective

Distributing manufacturing data across an enterprise requires the curation and management of the data within a repository and end-user services to access the data. The manufacturing data comes from a network of industrial equipment instrumented with sensors and other data-acquisition devices. Without a data-distribution system, data users must discover, curate, and integrate manufacturing data manually. The software system described in this SRS can enable the automation of the required tasks.

This SRS describes a software system that is part of a larger manufacturing-monitoring system. Figure 1 presents the architecture of the manufacturing-monitoring system. This SRS covers the VDS and QDR applications contained in the internal and external data access portions of the solution stack presented in Figure 1.

2.1.1 Volatile Data Stream

The VDS is an application that executes an MTConnect Agent, which is "a process that implements the MTConnect HyperText Transfer Protocol (HTTP) protocol, Extensible Markup Language (XML) generation, and MTConnect protocol" [1]. The VDS is a high priority for the software system.

MTConnect is an open-source, read-only, extensible data-exchange standard for manufacturing and was originally designed to transform process-related information from proprietary to structured-XML formats accessible for monitoring applications [1]. The standard is based on HTTP and provides information models and communication protocols to enhance the data-acquisition capabilities of manufacturing equipment, systems, and applications and enable a plug-and-play environment. Four information models are currently included in MTConnect: Devices, Streams, Assets, and Errors. These models are the only established common vocabulary and structure for manufacturing equipment data.

The VDS provides a means for streaming buffered data aggregated from the manufacturing devices registered in the system. The VDS must provide a report of all data items available in the VDS, a near-real-time stream of most current value for each data item, and a time series of most recent values collected for each data item available in the VDS buffer.

2.1.2 Query-able Data Repository

The QDR provides a means for transforming data into structured formats and sharing this structured data with users. The data are organized using predefined templates encoded in XML Schema Definition (XSD). These templates must be used to create data queries. The

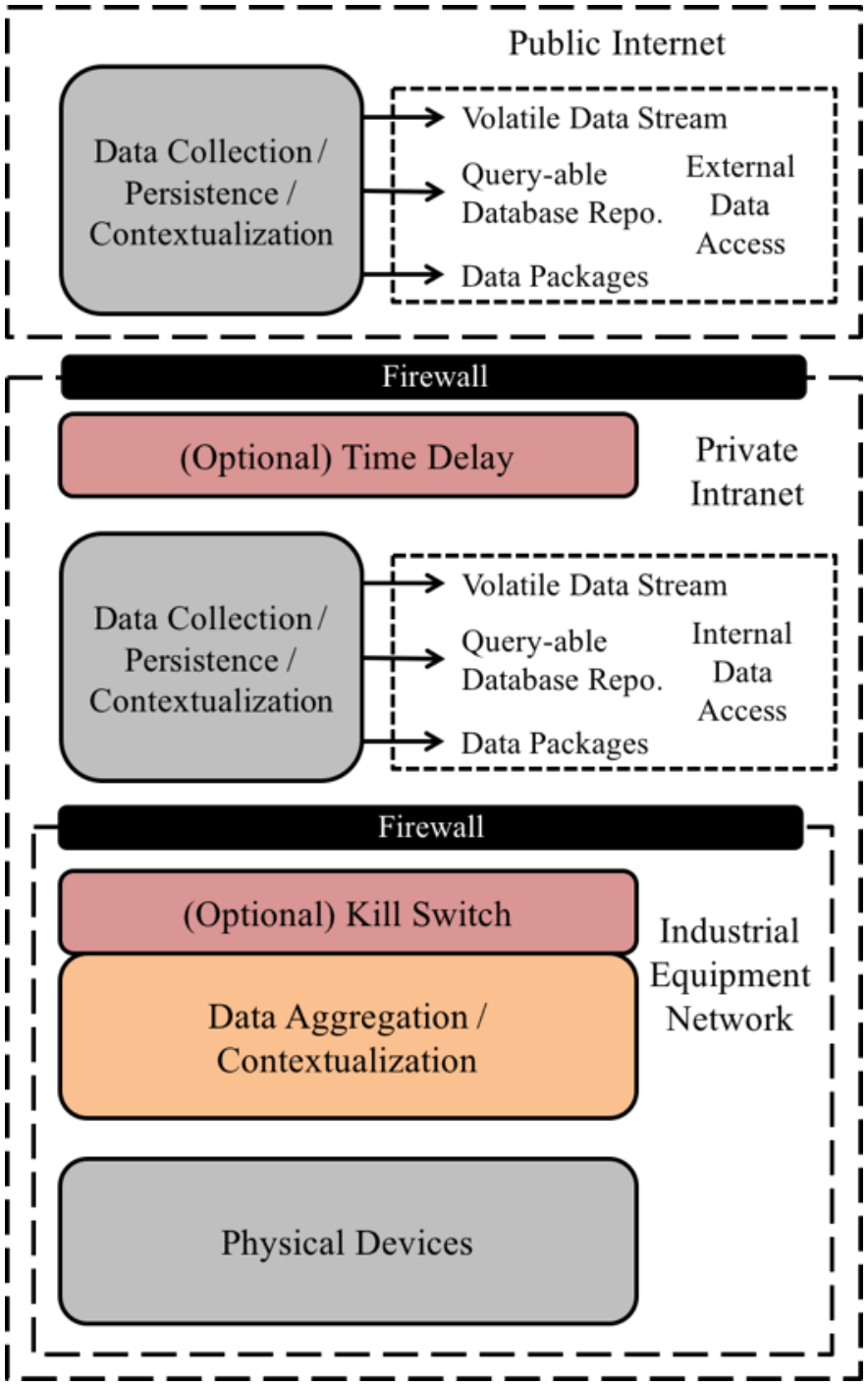

Fig. 1. Manufacturing-Monitoring System Stack

data must be saved. A Not only Structured Query Language (NoSQL) type of database or relational database may be used to save the data. The QDR user would search and retrieve data via a template-driven web-based form, an endpoint query, and/or a Representational State Transfer (REST)ful Application Programming Interface (API) call. The QDR is a high priority for the software system.

2.2 System Functions

The system must provide, at a minimum, the following functions in accordance with the other requirements described within this SRS document.

- Provide a data stream through the VDS in a form compliant with MTConnect and data templates in the QDR
- Upload, edit, and create data templates
- Make all data available over a RESTful API
- Provide a User Interface (UI) for administration and managing templates and queries
- Allow the administrator to create predefined user queries with limits set on results by a time range (e.g., number of days, hours, minutes), a limited set of query-able data elements, or a limited set of records
- List all data elements

2.3 User Classes and Characteristics

Users of the software system include researchers, practicing engineers, and software developers. All users should be familiar with web technology. Users are knowledgeable of software-development processes or manufacturing processes but may not be knowledgeable of both. However, users should have a good understanding of the tasks, activities, and artifacts of either process in which they may be interested.

2.4 Operating Environment

The server-side components of the software system must operate within a Linux operating system environment.

The client-side components of the software system must operate within common web-browser environments using Secure Sockets Layer (SSL) / Transport Layer Security (TLS) cryptographic protocols at a minimum encryption level of 128 bits. The minimum set of browsers that must be supported is:

- Apple Safari 7+
- Google Chrome 44+

- Microsoft Internet Explorer 10+
- Mozilla Firefox 40+

2.5 Design and Implementation Constraints

Developers may implement the QDR using a data store to curate data and template information. All QDR and administrative services must be available over a RESTful API. The QDR must be able to curate data collected in batches from the Industrial-Equipment Network. At a predetermined time each calendar day, the QDR must process all curated data and populate a list of available elements for query. Setup and maintenance of the QDR application are the responsibility of the customer. Setup and maintenance include: installation, hosting, host-security configuration, and administration. There are no additional constraints on the VDS other than any defined for an application that executes the MTConnect Agent in the standard documentation [1].

2.6 User Documentation

Any user of the software system is the target audience for user documentation generated about the software system. A range of short document types (e.g., guidelines, tutorials, frequently asked questions) in HyperText Markup Language (HTML) and/or Portable Document Format (PDF) format must describe the use of the software system.

2.7 Assumptions and Dependencies

No specific assumptions or dependencies are considered at this time.

3. External Interface Requirements

This section of the SRS describes the interface requirements for the system. Requirements for user, hardware, software, and communication interfaces are defined.

3.1 User Interfaces

This subsection of the SRS defines the UI requirements for the VDS and QDR.

3.1.1 Volatile Data Stream

The UI for the VDS must comply with the requirements of the application that executes the MTConnect Agent as defined by the standard documentation [1].

3.1.2 Query-able Data Repository

All pages of the QDR UI should employ an organizational method that separates the major sections of the application for easy access. The user should be able to determine the sections of the QDR UI currently being viewed using visual cues, such as bold fonts, arrows, and highlighting. The application's major sections should be available from any page of the QDR and include the following pages: Query, Help, Introduction, Administration, Login, and Logout.

The user must be presented with a login page when accessing any page, expect a landing or front page, as an anonymous user. Administrative tasks and running queries must have a UI. The UI for administrative tasks must included the ability to manage Users, manage Groups, manage permissions, manage data templates, manage group assignments, and manage query templates. The UI for running queries may include the ability to query the database using query templates defined by the administrator. All pages should use a consistent visual theme. Pop-ups displaying notifications, error messages, or warnings should use the same theme and template as the rest of the application.

3.2 Hardware Interfaces

All server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

3.3 Software Interfaces

The software interface should follow the Model-View-Controller (MVC) model for rendering and modeling data objects. The interface must be able to connect to a database to store XML schema defined using XSD and data streams. Source and destination formats for data must include XML and may also include: Extensible Stylesheet Lanaguage Transformation (XSLT), JavaScript Object Notation (JSON), Comma Separated Value (CSV), and

American Standard Code for Information Interchange (ASCII).

3.3.1 Description of Services

Data templates act as intermediate format between the API, UI, and Database.

Administration interfaces with: Database, REST API, and UI.

- Input: Create, Retrieve, Update, and Delete (CRUD) information
- Output: CRUD administrative data
- Relationships: Users, Groups, Permissions, Templates

VDS Capture and Data Curation interface with: Database, REST API, and MTConnect Agent.

- Input: Machine adapter stream
- Output: MTConnect compliant XML
- Relationships: Templates

QDR Capture or Data Curation interfaces with: Database and REST API.

- Input: XML document conforming to pre-defined QDR template (e.g., Quality Information Framework (QIF) schema, MTConnect schema)
- Output: Stored data in QDR database
- Relationships: Templates

QDR Template Composer interfaces with: Database and UI.

- Input: Template information
- Output: XML template
- Relationships: XML data, database, data view

QDR Query interfaces with: Database, REST API, and UI.

- Input: Query parameters
- Output: Query results
- Relationships: Curated data, time series

Client-side UI applications must run within a web browser and not require any additional software interfaces on the client.

3.4 Communication Interfaces

The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP Secure (HTTPS). The client-server communication must be stateless. A uniform interface must separate the client roles from the server roles. Communication over the VDS must conform to a reference MTConnect data schema [2] using XML. Communication over the QDR must provide the user with the option of formatted ASCII CSV, JSON, or XML structured with a reference MTConnect data schema [2].

4. System Features

This section of the SRS describes the requirements for the system's features. Specifically, requirements for component functionality, data curation, and software system administration are defined.

4.1 Component Functional Requirements

This subsection of the SRS describes the functional requirements for the VDS and QDR components of the software system.

4.1.1 Volatile Data Stream

VDS/001 The VDS component of the software system must comply with all enumerated requirements for an application to execute an MTConnect Agent in the standard documentation [1].

4.1.2 Query-able Data Repository

QDR/001 The QDR component of the software system must retrieve data from an indexed database based on a user-defined query request via a template-driven web-based user interface, an endpoint query, and/or a RESTful API call.

QDR/002 The QDR must provide the administrator with the ability to predefine queries for use by non-administrator users.

QDR/003 The QDR must enable the administrator to limit query results by date ranges and/or number of records.

QDR/004 When configured in a query by the administrator, the QDR must provide the user a list of all available devices in the database and all available associated data elements for that device.

QDR/005 The user may be provided the ability to select what data elements to retrieve via the predefined query.

QDR/006 The user must have the ability to select the time range and/or number of records for the data to retrieve via the query.

QDR/007 The user must have the ability to retrieve and view data as formatted ASCII CSV, JSON, or XML.

QDR/008 The user must have the ability to download retrieved data as ASCII CSV, JSON, or XML.

4.2 Data Curation

This subsection of the SRS describes the requirements for data curation within the software system.

4.2.1 Description and Priority

The software system must provide a function to store and manage data in a database according to a data-structure template (e.g., schema, ontology).

4.2.2 Functional Requirements

Curate/001 The software system must provide a RESTful API for curating data in the database.

Curate/002 The software system must provide the data curator with the ability to add data to the database in accordance with a stored data-structure template.

Curate/003 The software system may provide all curation functionality via a web-based user interface.

4.3 Software System Administration

This subsection of the SRS describes the requirements for administering the software system.

4.3.1 Description and Priority

The software system must provide a function to manage user accounts and data-structure templates (e.g., schemas, ontologies). The minimum user account types must be: (1) data explorer, (2) data curator, and (3) administrator. The data-explorer account type will grant access to end-users for the purpose of interacting with the QDR component of the software system. The data-curator account type will grant access to a user for the purpose of storing and managing data in the database. The administrator account type will grant "super user" access to a user for the purpose of managing the software system. The administration function of the software system is a medium priority.

4.3.2 Functional Requirements

Admin/001 The software system must provide a web-based UI to CRUD manage users.

Admin/002 The software system must provide the administrator with the ability to CRUD manage users.

Admin/003 The software system must provide the administrator with the ability to CRUD manage groups.

Admin/004 The software system must provide the administrator with the ability to CRUD manage user group assignments.

Admin/005 The software system must provide the administrator with the ability to CRUD manage permissions assigned to users and groups.

Admin/006 The software system must provide a web-based user interface for CRUD manage data templates.

Admin/007 The software system must provide the administrator with the ability to CRUD manage data-structure templates using the XSD format.

Admin/008 The software system may provide all administration functionality via a RESTful API.

Admin/009 The software system must allow the administrator to create a Universally Unique Identifier (UUID) associated with a device.

Admin/010 The software system must require a UUID to be provided with any data streams uploaded.

5. Non-functional Requirements

This section of the SRS describes the non-functional requirements for the software system. Performance, reliability, availability, security, and maintainability requirements are included. In addition, business rules are defined.

5.1 Performance Requirements

The VDS must provide available fresh data to the requesting client at a rate of no less than 1 Hz. The response-time for the QDR should be minimized using industry recommended practices. All other performance related to storage, memory, and processing should follow industry recommended practices to ensure resource requirements are minimized.

5.2 Reliability Requirements

There are no requirements for Mean Time between Failures (MTBF) or Mean Time to Failures (MTTF) for this version of the software system defined in this SRS. However, in accordance with industry recommended practices, the software system should undergo feature testing, load testing, and regression testing prior to release and/or deployment.

5.3 Availability Requirements

Reasonable efforts should be made to ensure the software system is available with an uptime of 95%. The uptime is calculated as the percentage of time during the year in which the software system was available to the public. A 95% uptime percentage allows for an average of 18.25 days per year, 36 hours per month, or 8.4 hours per week of downtime.

5.4 Security Requirements

The software system defined in this SRS must follow industry recommended practices for secure software development. At a minimum, the software development must practice the principle of least privilege for defining access-level requirements of the software system and its associated services. The production-release version of the software system must pass an automated dynamic application security testing tool (e.g., HP WEBINSPECT).

5.5 Maintainability Requirements

The architecture, design, implementation, and documentation of the application must minimize the maintenance costs of the software system. The maximum person-time required to fix a security defect (including regression testing and documentation update) must not exceed two person days. Otherwise, the software system must be taken offline or the offending feature disabled. The average person-time required to make a minor enhancement (including testing and documentation update) must not exceed one person week.

5.6 Business Rules

The system must have at least a *Super-User* role and a *User* role defined for accessing and interacting with the system. Additional roles may be defined for the system as long as the business rules for the administrator and user roles are satisfied. At a minimum, the Super-User role must account for the *data explorer*, *data curator*, and *administrator* account-type requirements described in Section 4.3. At a minimum, the User role must account for the *data explorer* account-type requirements described in Section 4.3. The following business rules must apply to the administrator and user roles.

5.6.1 Super-User Role Business Rules

SuperUserRole/001 Maintains all VDS and QDR back-end system configurations

SuperUserRole/002 Maintains all VDS and QDR schemas and templates

SuperUserRole/003 Maintains all user groups and user accounts

SuperUserRole/004 Maintains all QDR predefined queries

5.6.2 User Role Business Rules

UserRole/001 Connects to and retrieves and reads data from VDS and QDR in conformance with the maintained VDS and QDR schemas and templates

Appendix A: Glossary

Application Programming Interface (API) A system of tools and resources (e.g., subroutine definitions, protocols) in an operating system that enables developers to create software applications.

Extensible Markup Language (XML) A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable in accordance with the World Wide Web Consortium (W3C) specification.

Extensible Stylesheet Lanaguage Transformation (XSLT) A language for transforming XML documents into other XML documents or other formats in accordance with the World Wide Web Consortium (W3C) specification.

HyperText Markup Language (HTML) The standard markup language for creating web pages and web applications in accordance with ISO/IEC 15445 and W3C HTML5.

HyperText Transfer Protocol (HTTP) An application protocol for distributed, collaborative, and hypermedia information systems in accordance with RFC 7230 and RFC 7540.

HyperText Transfer Protocol Secure (HTTPS) A communications protocol for secure communication over a computer network using Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL).

JavaScript Object Notation (JSON) An open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types.

Mean Time between Failures (MTBF) The predicted average elapsed time between inherent failures of a system during operation.

Mean Time to Failures (MTTF) The predicted average operation time for a system replaced after a failure.

MTConnect An open industry standard for enabling interoperability between controls, devices, and software in manufacturing systems.

Not only Structured Query Language (NoSQL) Database A database providing a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

Quality Information Framework (QIF) A unified XML framework and ASNI-accredited standard for enabling interoperability between computer-aided quality measurement systems.

Representational State Transfer (REST) A style of architecture based on a set of principles that describe how networked resources are defined and addressed for providing interoperability between computer systems on the Internet.

Secure Sockets Layer (SSL) An early standard security technology, later replaced by Transport Layer Security (TLS), for establishing an encrypted link between a web server and a browser.

Transport Layer Security (TLS) A cryptographic protocols that provides communications security over a computer network in accordance with RFC 6176.

Universally Unique Identifier (UUID) A 128-bit number generated in accordance with ISO/IEC 11578 used to identify information in computer systems. Also known as Globally Unique Identifier (GUID).

XML Schema Definition (XSD) A schema language recommended by the World Wide Web Consortium (W3C) for expressing a set of rules to which an XML document must conform to be considered valid.

Appendix B: Reference Diagrams and Models

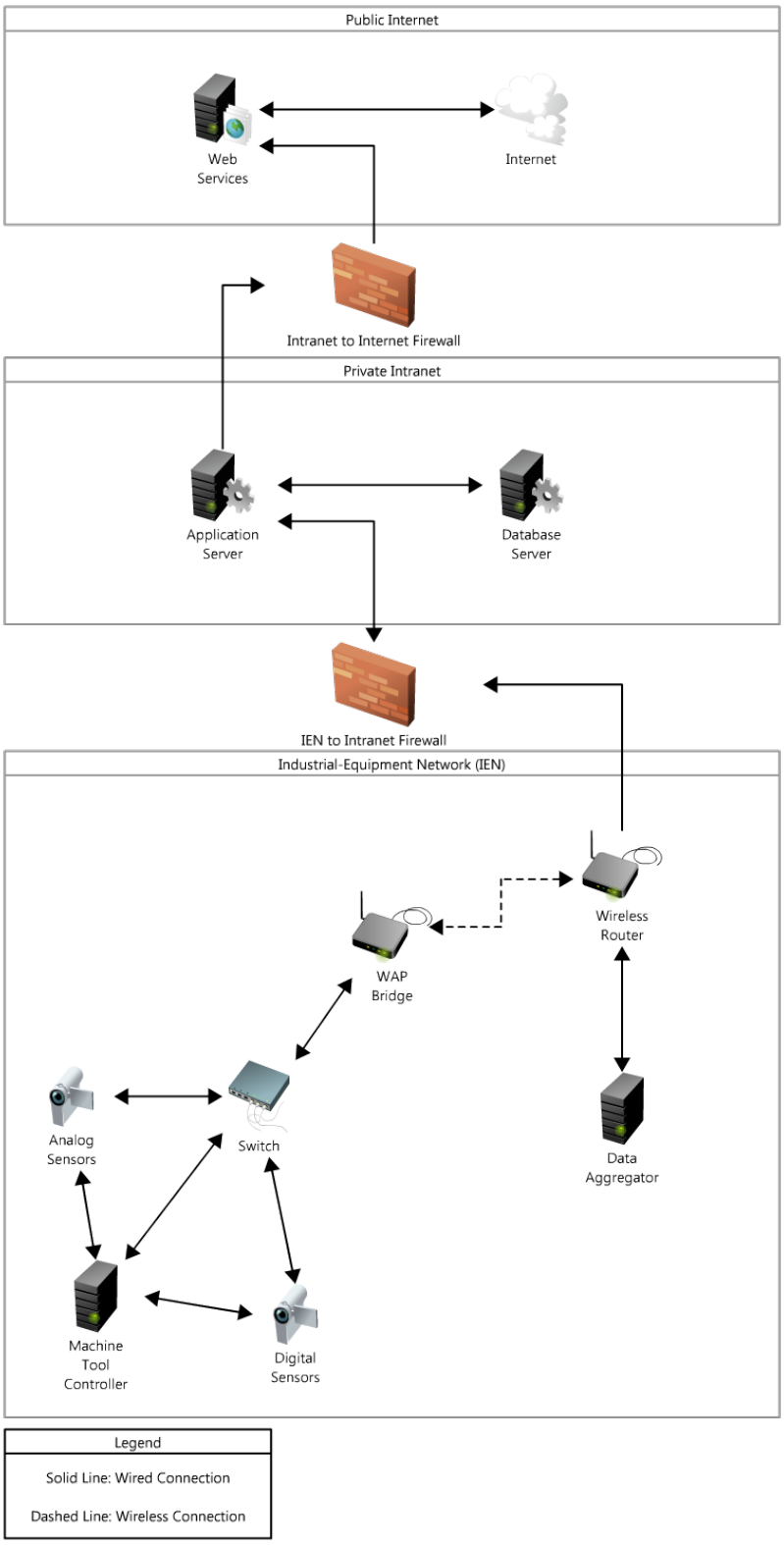

Fig. 2. Example network diagram for a smart manufacturing system design to collect, curate, and disseminate manufacturing data.