

Published in final edited form as:

CIRP J Manuf Sci Technol. 2017 ; 19: . doi:10.1016/j.cirpj.2017.04.002.

Reference architecture to integrate heterogeneous manufacturing systems for the digital thread

Moneer Helu, Thomas Hedberg Jr., and Allison Barnard Feeney

Engineering Laboratory, National Institute of Standards and Technology (NIST), 100 Bureau Drive, Gaithersburg, MD 20899, USA

Abstract

The increasing growth of digital technologies in manufacturing has provided industry with opportunities to improve its productivity and operations. One such opportunity is the digital thread, which links product lifecycle systems so that shared data may be used to improve design and manufacturing processes. The development of the digital thread has been challenged by the inherent difficulty of aggregating and applying context to data from heterogeneous systems across the product lifecycle. This paper presents a reference four-tiered architecture designed to manage the data generated by manufacturing systems for the digital thread. The architecture provides segregated access to internal and external clients, which protects intellectual property and other sensitive information, and enables the fusion of manufacturing and other product lifecycle data. We have implemented the architecture with a contract manufacturer and used it to generate knowledge and identify performance improvement opportunities that would otherwise be unobservable to a manufacturing decision maker.

Keywords

Data architecture; Digital thread; Systems integration

1. Introduction

Digital technologies and solutions have grown tremendously in manufacturing. For example, only 8 % of firms in 1999 used data warehouses to support their operations [1]. Today, industry has become focused on concepts like smart manufacturing, Industrie 4.0, and cyber-physical systems [2–4]. The steady adoption of data standards, such as MTConnect [5] and OPC Unified Architecture (OPC UA) [5], have enabled a growing market of digital solutions for data-driven, web-enabled manufacturing [4]. The increasing accessibility and growth of these standards and technologies has provided industry with opportunities to leverage data to reduce costs, improve productivity, ensure first-pass success, and augment existing workforce capabilities [2,4]. These opportunities can also address evolving industry challenges created by the increasingly distributed nature and growing complexity of modern manufacturing systems and global production networks [4].

Disclaimer: Identification of commercial systems does not imply recommendation or endorsement by NIST or that the products identified are necessarily the best available for the purpose.

One opportunity enabled by digital technologies that has been gaining attention is the “digital thread” concept. The digital thread links disparate systems across the product lifecycle and throughout the supply chain [7]. It enables the collection, transmission, and sharing of data and information between systems across the product lifecycle quickly, reliably, and safely. This concept drives data-driven applications that can generate domain-specific knowledge for decision support, requirements management, and ultimately improved diagnosis, prognosis, and control of design and manufacturing processes. Such capabilities address a significant need for manufacturing where traditional decision-making and paper-based processes often neglect the far-reaching implications of specific actions on the product lifecycle.

The development of the digital thread concept (and in fact many digital technologies) in manufacturing has been challenged by the inherent difficulty of aggregating and applying context to data from systems across the product lifecycle [4,7–9]. These systems generate and require various types of data of different formats stored using different means in different locations [7,9]. Commercial systems exist that attempt to organize product lifecycle data, but these systems often lock users into homogeneous suites of solutions throughout the enterprise. This can add additional expense onto products that are already extremely expensive and out of reach for many organizations, especially small-to-medium enterprises (SMEs). These solutions also do not often address the “silo effect” between engineering and manufacturing functions or between different organizations across the supply chain. There is a strong need for data infrastructures and management concepts that are scalable, integrate with heterogeneous systems, cut across many domains, and enable industry to determine where best to leverage data. The goal of this paper is to describe our development and implementation of a reference architecture to enable the digital thread in manufacturing.

2. Background

Much of the research on the curation and use of data for decision support has focused on the development and implementation of applications rather than the management of data itself [1]. The lack of well-established architectures for digital manufacturing has limited the use of decision-support systems since an application cannot successfully generate knowledge without appropriately managing the flow and contextualization of data. Addressing this need of industry is especially important because traditional architectures in manufacturing have been challenged by the growing use of digital standards and technologies [4].

2.1. Traditional Architectures in Manufacturing

There is no single unified data architecture that is used across all industrial sectors of manufacturing [3,10]. Instead, firms have deployed one-, two-, and three-tier architectures to address the various use cases that each has faced [3]. An n-tier (or multilayer) architecture approach describes the separation and modularization of capabilities in a computing environment. These capabilities are referred to as logic mechanisms, which manage application commands, logical decisions, and computations as data moves between layers of the architecture. Separating logic mechanisms provides developers and users with flexibility when implementing and maintaining solutions to address needs and requirements.

Specifically, developers and users do not need to develop new or redevelop entire applications whenever change is required. The primary goal of the n-tier architecture approach is to enable solutions within each tier that are specialized to a specific task needed to manage, contextualize, and present data.

Figure 1 provides an overview of common one-, two-, and three-tier architectures. Systems within the lowest tier in each architecture (labelled “Data”) curate the data that is to be accessed and potentially used for analysis. Systems within the highest tier in each architecture (labelled “Client”) consume the data to provide knowledge to a user or external system. Systems within the middle tiers in each architecture provide services that manage the transformation, translation, and/or transaction of data between curators and consumers.

A typical one-tier architecture in manufacturing is composed of individual workstations, such as manufacturing equipment or computers, that both provide and consume data and information. There is little to no connectivity between these systems. For example, part programs may be generated on-machine or via computer-aided manufacturing (CAM) software on a computer and then copied to the machine using a storage device (e.g., floppy disk, memory stick). Such solutions are colloquially called “sneaker net,” which describes the manual process of copying files from one system to another. One-tier architectures silo manufacturing systems and prevents these systems from coordinating activities.

Two-tier architectures begin to address the issues of one-tier architectures described previously. An example of a two-tier architecture in manufacturing is composed of a file or application server (e.g., to host a CAM license and generate G-code to control a machine tool) and client (e.g., the machine tool that consumes and uses the G-code). Each client typically exists on a network that enables it to communicate directly with databases or servers in Tier 2. Most logic mechanisms exist within the client, which can create additional computing burdens on these clients. While two-tier architectures provide some value (e.g., hosting part programs in a centrally-managed location), they do not usually provide sensing, monitoring, and control mechanisms outside of basic human observation. Any data that is collected must be extracted manually from the manufacturing system and entered manually into a server via a client terminal, which makes the use of this data for data-driven applications cumbersome and error prone.

Three-tier architectures build upon two-tier architectures by beginning to separate and modularize the functions and roles in information technology (IT) and operational technology (OT) systems. A typical three-tier architecture in manufacturing is composed of a database, application server, and thick client (i.e., dedicated software applications used to access other tiers, such as the user interface of SAP or Siemens Teamcenter). The application server provides the interface between the database and client by giving the client access to business logic mechanisms, which generate dynamic content from the data in Tier 1. By removing these logic mechanisms from the client, a three-tier architecture simplifies the design and use of the client and eases IT maintenance requirements and functionality upgrades. Three-tier architectures in manufacturing have given rise to product data management (PDM), manufacturing execution system (MES), and enterprise resource planning (ERP) solutions [11].

In the same way that IT systems in other domains have progressed from one- to two- to three-tier architectures, traditional architectures in manufacturing have evolved to better manage the data collected to support production operations. Despite this evolution, though, one-, two-, and three-tier architectures co-exist currently across industry. This lack of consistency has challenged the development of digital manufacturing because different types of content have different requirements for storage, processing, service, and observation [11]. These requirements have forced organizations to deploy different mechanisms and systems for each content type. This situation is also partly due to the lack of vertical integration between systems from the shop floor to the operations and enterprise levels [12]. The International Society for Automation (ISA) 95 standard [13] defines manufacturing systems as being composed of five levels: physical processes (Level 0), sensing and manipulating (Level 1), monitoring and control (Level 2), workflow and operations (Level 3), and business planning and logistics (Level 4). Typically, Level 1 comprises sensors and actuators, Level 2 comprises supervisory control and data acquisition (SCADA) systems and human-machine interfaces (HMI), Level 3 comprises MES, and Level 4 comprises ERP. The lack of integration across the five levels of ISA 95 has led to the manual or semi-automated duplication of data across all systems within the manufacturing enterprise [12].

2.2. Moving to Extended Enterprises: Four-Tier Architectures

A goal of digital, data-driven manufacturing is for manufacturers to access their systems and the data being generated by their systems as they would access the Internet (e.g., through web browsers, dedicated software and mobile applications, and microservices) [4,7]. This goal has driven the need for an architecture that can support multiple clients that each may have different viewpoints and capabilities. Supporting multiple types of clients is a challenge for a three-tier architecture because different presentation logic mechanisms (i.e., software that provides presentation and representation of dynamically-generated content) are needed for each type of client [11]. The need to separate presentation and business logic mechanisms has led to the development of a four-tier architecture (see Figure 2); the aggregation tier (Tier 2) contains business logic mechanisms while the delivery tier (Tier 3) contains presentation logic mechanisms. Thus, various types of clients may be supported without having to redevelop the entire middleware layer for each type of client.

Four-tier architectures gained prominence in the mid to late 2000s after the rise of web services and service-oriented architectures (SOA). SOAs support four-tier architectures by enabling each component within the four-tier architecture to have public interfaces specific to the component and through which interactions with other components may occur [12]. By implementing web services and SOAs on all tiers, the four-tier architecture has improved flexibility and modularity since systems may be exchanged in and out of the four-tier architecture with little to no disruption to the other components of the architecture.

Four-tier architectures have become increasingly used for applications on the Internet and World Wide Web within ecommerce, reservation systems, and applied research. For example, the European Organization for Nuclear Research (CERN) has used a four-tier architecture to enable scientists to discover and interact with the millions of gigabytes of data generated by all of CERN's research activities [14]. CERN has thousands of data

systems and applications, each with different types of content and format. It would be challenging (if not impossible) to generate one-to-one integrations between all of the different systems used at CERN and then train all users on each system. By integrating their systems using a four-tier architecture, CERN allowed the data to be curated once, aggregated using linked data and business logic in application servers, and presented to user clients using various presentation-focused logic rules.

The CERN use case is analogous to integrating production systems vertically across all ISA 95 levels within the manufacturing stage and horizontally across multiple stages of the product lifecycle. This vertical and horizontal integration of production systems is called the “Extended Enterprise.” Adopting a four-tier architecture for this integration would ease and focus implementation on stakeholder requirements rather than specific integration details that must be repeated for every new system. Such an architecture would also provide a flexible, reliable integration environment that provides near plug-and-play capability. In this research, we develop and implement a reference four-tier architecture that integrates systems within the manufacturing domain as a first step to manage, contextualize, and share manufacturing data effectively across the digital thread.

3. Design and Implementation of the Reference Architecture

The systems integration process began by clearly describing the use case so that the correct set of specifications and requirements based on stakeholder needs and other relevant constraints could be identified and defined. The use case for this research was the collection and management of data from the design, planning, manufacturing, and inspection stages of the product lifecycle to support engineering change requests (ECRs) and dynamic scheduling and process control. The data in each stage was collected using generally accepted industry standards where possible. The types of data available included:

- *Design*: Standard for the Exchange of Product Data (STEP) and native computer-aided design (CAD) formats
- *Planning*: ISO 6983 (G code) and native CAM formats
- *Manufacturing*: MTConnect
- *Inspection*: Quality Information Framework (QIF)

Our goal was to link each type of data so that discrepancies may be identified, monitored, and diagnosed. The ultimate goal of this effort was to be able to build knowledge so that discrepancies can be predicted and corrected before they occur.

It is critical that the appropriate context be created when linking different types of data [8]. Without the appropriate context, it is impossible to extract high-quality knowledge that may be used effectively for decision support, control, or any other purpose. Often, the appropriate context can be created by simply mapping all relevant data sets back to some aspect that controls the phenomenon of interest. For example, many manufacturing studies require that measurements map to process parameters via common timestamps so that one can identify the physical reasons causing observed variations in the measurements.

The computing environment for much of the manufacturing equipment was another important aspect of our use case. Manufacturing equipment tends to use outdated operating systems that represent a significant cybersecurity risk when networked. It is critical that these systems be isolated as much as possible to prevent external actors from taking control of the equipment and/or obtaining sensitive information (e.g., intellectual property about the part being fabricated or the process being used to fabricate the part).

A final important set of considerations was the type of client interactions we anticipated for users of the generated data. We decided to provide three means of access to the data: (1) volatile data stream; (2) query-able database repository; and (3) data packages. The volatile data stream is the near-real-time stream of data flowing from each component of the system that can be used for monitoring, control, and other applications. The query-able database repository is the historically collected data from the system that can be queried to support specific research goals. The data packages are resources to be developed for end users to verify and validate the conformance of their systems to existing standards and best practices.

The remaining requirements and specifications that were generated from our use case were documented by Hedberg et al [15]. While too numerous to describe completely in this paper, areas covered by the specifications and requirements included interfaces (e.g., users, hardware, software, communications), features (e.g., data curation, system administration), and performance (e.g., reliability, availability, security). Based on these requirements, we developed the architecture shown in Figure 3 to manage manufacturing (i.e., MTConnect) data for our use case.

The reference architecture is designed across three networks: (1) industrial equipment network; (2) private intranet; and (3) public internet. The industrial equipment network shields physical devices from external actors. It only allows data to be pushed from inside the network to the *private* intranet, which provides further protection. The use of a private intranet and a *public* internet provides segregated access to internal and external clients, which enables an organization to share only necessary information with suppliers and partners.

Again, the reference architecture has been designed as a four-tier architecture that follows the structure described in Figure 2. Tier 1 (Data) represents the Physical Devices that serve as the source of manufacturing data. Tier 2 (Aggregation) represents Data Aggregation and Contextualization where different data streams for each component are aggregated and contextualized relative to the process or activity occurring at the component. These components also provide data protocol translation and supply the structure of the data for underlying services. Tier 3 (Delivery) represents the Data Collection, Persistence, and Contextualization that occurs in both the private intranet and public internet. Here, data is processed for delivery to the client and content is cached for efficient performance. It also enables further development through data analytics (e.g., identifying the types of data accessed most to inform what data may be most valuable to collect for different questions and decisions). Finally, Tier 4 (Client) represents the user interface that provides the three means of data delivery described previously (i.e., volatile data stream, query-able database repository, and data packages).

The reference architecture described in Figure 3 has been implemented with a contract manufacturing partner through the National Institute of Standards and Technology (NIST) Smart Manufacturing Systems Test Bed [16]. Tier 1 (Services) includes all networked shop-floor IT and OT systems, such as machine controllers, external sensors deployed on each machine, inspection systems, and production management systems. Tier 2 (Aggregation) includes two pieces of software per device: (1) an adapter to translate each data stream into the MTConnect standard and (2) an MTConnect agent to aggregate all data streams for one device and provide a common timestamp. Tier 3 (Delivery) includes one MTConnect agent that aggregates the data from every device agent in Tier 2 and parses XML documents for delivery to the query-able database repository. Finally, Tier 4 (Client) consists of web applications, including a replicated MTConnect agent from Tier 3 for the volatile data stream and web interfaces to provide access to the query-able database repository and data packages. Data moves from one tier to the next tier using custom scripts programmed to automate the data transfer on a daily basis. The data itself is transferred as simple text files with 10000 data points (typically about 1.5 MB in size) formatted based on the output of a typical MTConnect adapter.

4. Case Study and Summary

To explore the opportunities that can be enabled by the reference architecture, we fabricated a production part provided by an industrial partner. Using our implementation, we collected data on the as-manufactured state of the part using MTConnect and compared it to a cutting simulation generated from the ISO 6983 (G code) data. The measured cycle time was twice as long as the simulation due mainly to one portion of the toolpath with significant feedrate mismatch. By layering data from design (STEP), planning (ISO 6983), and manufacturing (MTConnect), we were able to determine that the feedrate mismatch was due to one feature machined using a zig-zag toolpath that required many quick changes in acceleration. Further investigation uncovered that the design of this feature was a legacy artifact no longer needed for this part. This performance improvement opportunity would not have been realized without the fused, contextualized data enabled by the reference architecture.

As our case study shows, the reference architecture generated by this research supports the vertical integration of different manufacturing systems so that the data from these systems may be effectively managed, contextualized, and used to generate knowledge and value for the decision maker. Using a four-tier architecture enables seamless vertical and horizontal integration across different product lifecycle stages (e.g., design, planning, manufacturing, inspection) without the need for larger homogenous systems typical of modern product lifecycle management (PLM) solutions. Experts within each lifecycle stage can focus on vertically integrating systems in their domain so that data can be collected at the lowest levels, aggregated up through the architecture, and presented to a user or client application. The aggregation and delivery tiers from one domain can interface with systems in other lifecycle stages using application programming interfaces (APIs), semantic web, and linked data methods. The resulting extended enterprise solution would enable effective and efficient synthesis of data across the lifecycle, which would support better decision making and improved control.

The goal for future work is to automate the data aggregation and contextualization further as well as expand the reference architecture to other lifecycle stages outside of manufacturing to provide a complete perspective of the product lifecycle. Doing so has the potential to provide tremendous value to manufacturers. As the case study highlighted, the context created by simply linking product lifecycle data has the potential to diagnose root causes that are typically hidden and enable effective control of design and manufacturing processes that achieves the promise of digital manufacturing.

References

- [1]. Bhargava HK, Power DJ, Sun D. (2007) Progress in web-based decision support technologies. *Decision Support Systems*, 43:1083–1095.
- [2]. Evans PC, Annunziata M. (2012) Industrial internet: pushing the boundaries of minds and machines. General Electric.
- [3]. Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, Helu M. (2015) Cloud-enabled prognosis for manufacturing. *CIRP Annals*, 64(2):749–772.
- [4]. Monostori L, Kadar B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, Sauer O, Schuh G, Shin W, Ueda K. (2016) Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2):621–641.
- [5]. MTConnect Institute. (2015) MTConnect v. 1.3.1.
- [6]. OPC Foundation. (2017) OPC Unified Architecture.
- [7]. Hedberg T Jr, Barnard Feeney A, Helu M, Camelio JA. (2016) Towards a lifecycle information framework and technology in manufacturing. *J. Computing and Information Science in Engineering*, doi: 10.1115/1.4034132.
- [8]. Laroche F, Dhuieb MA, Belkadi F, Bernard A. (2016) Accessing enterprise knowledge: a context-based approach. *CIRP Annals*, 65(2):189–192.
- [9]. Colledani M, Tolio T, Fischer A, Iung B, Lanza G, Schmitt R, Vancza J. (2014). Design and management of manufacturing systems for production quality. *CIRP Annals*, 63(2):773–796.
- [10]. Dassisti M, Panetto H, Tursi A. (2006) Product-driven enterprise interoperability for manufacturing systems integration Proc. Business Process Management 2006 International Workshops, Vienna. Springer, 249–260.
- [11]. Behravanfar R (2001) Separation of data and presentation for the next generation Internet using the four-tier architecture. Proc. 39th Intl. Conf. and Exhibition on Technology of Object-Oriented Languages and Systems, 83–88.
- [12]. Al Azevedo, Toscano C (2000) An information system for distributed manufacturing enterprises. *Enterprise Information Systems*. Springer, 131–138.
- [13]. Intl. Society of Automation. (2010) Enterprise-Control System Integration – Part 1: Models and Terminology.
- [14]. Bray R (2015) Complementing complex PLM systems with simplified user interfaces. 3D Collaboration and Interoperability Congress, Chantilly, VA.
- [15]. Hedberg T Jr, Helu M, Newrock M. (2017) Software requirements specification to distribute manufacturing data. NIST Advanced Manufacturing Series 300–1.
- [16]. Helu M, Hedberg T Jr. (2015) Enabling smart manufacturing research and development using a product lifecycle test bed. *Procedia Manufacturing*, 1:86–97. [PubMed: 28664167]

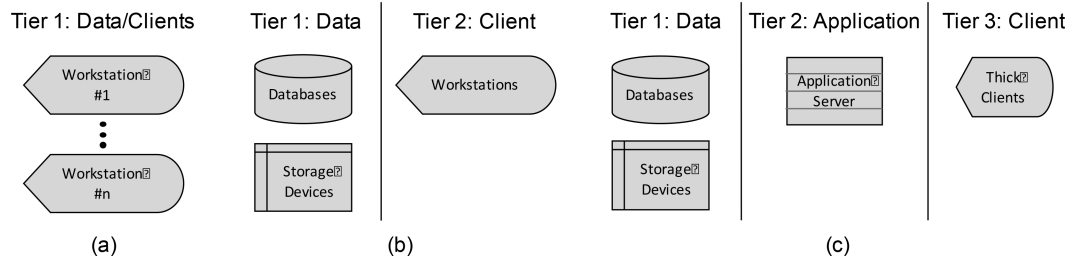


Figure 1. Schematic examples of n-tier (or multilayer) architectures used in manufacturing: (a) one-tier; (b) two-tier; and (c) three-tier.

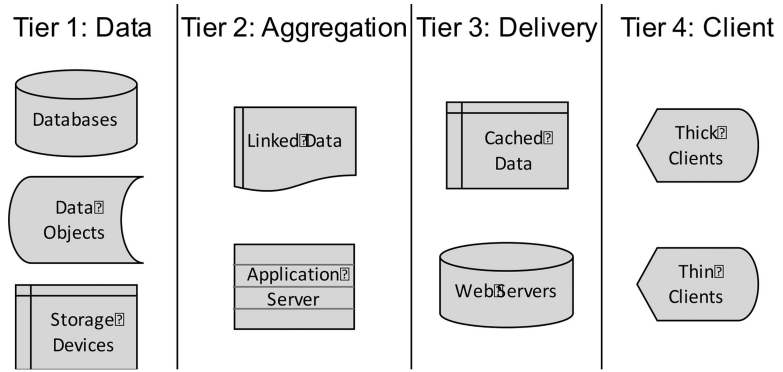


Figure 2. Schematic example of a four-tier architecture.

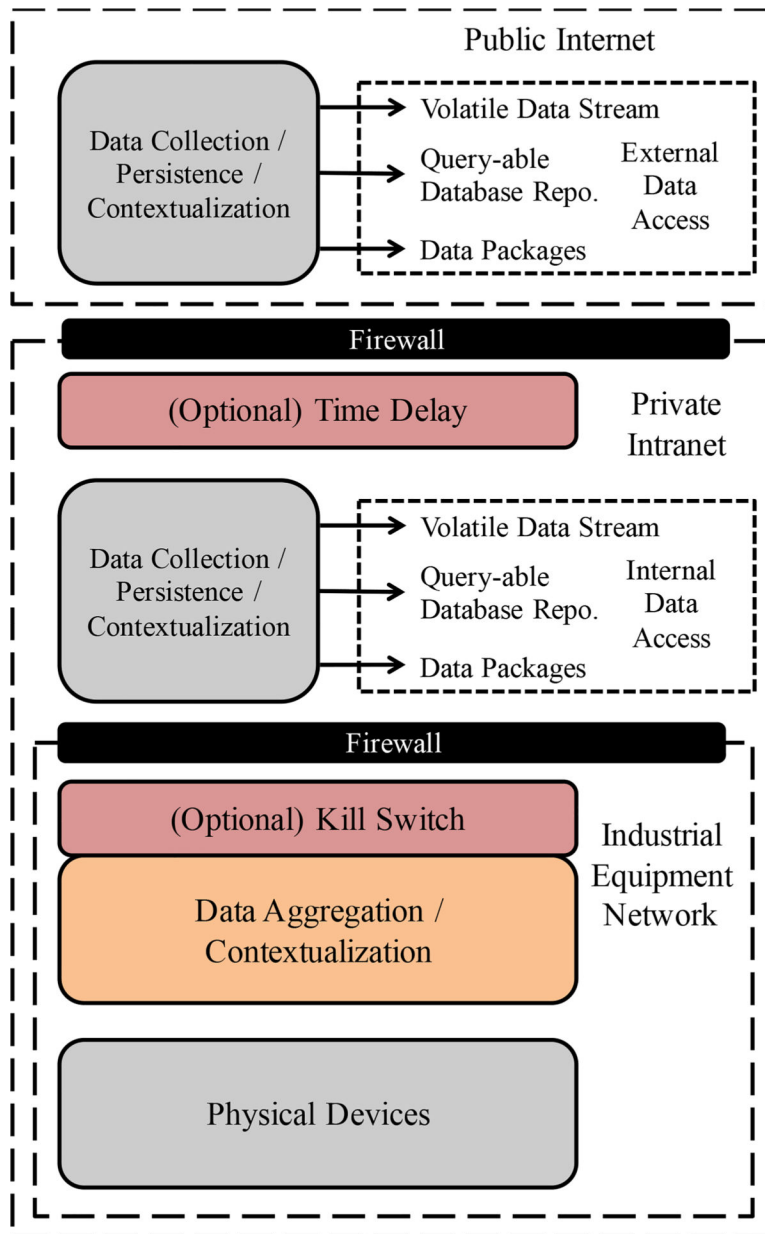


Figure 3. Reference architecture to manage manufacturing data.