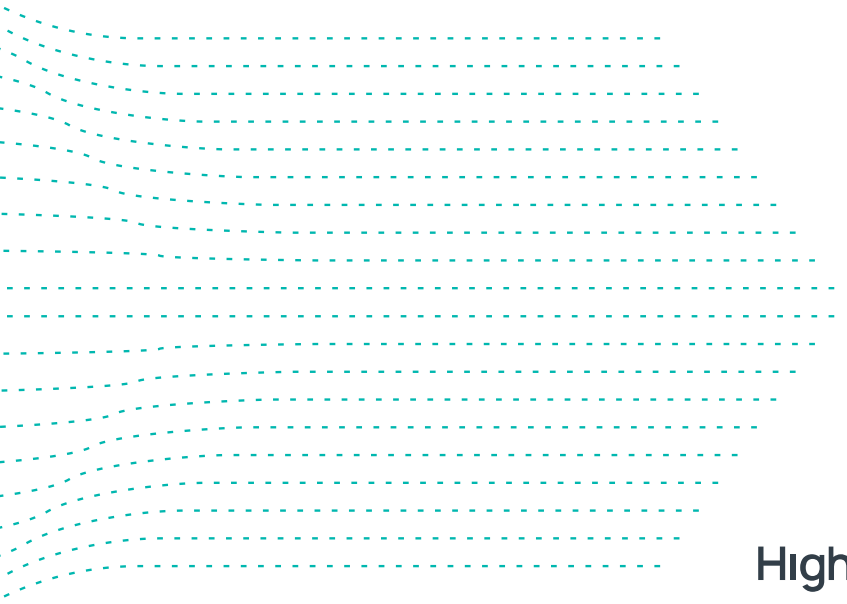
Novotek

Think Big, Start Small, Scale Fast: The Data Engineering Workbook

10 steps to designing an industrial data architecture for scale



HighByte

Introduction

Many manufacturers are drowning in data and struggling to make it useful. A modern industrial facility can easily produce more than a terabyte of data each day. With a wave of new technologies for artificial intelligence and machine learning coupled with real-time dashboards, we should see huge gains in productivity. Unplanned asset and production line maintenance should be a thing of the past, but even today, that is not the case.

Access to data does not make it useful. Industrial data is raw and must be made fit for purpose to extract its true value. Furthermore, the tools used to make the data fit for purpose must operate at the scale of an industrial enterprise.

With these realities in mind, here is a practical, 10-step guide with a few short questionnaires to help manufacturing and industrial leaders better understand the process of applying proven data engineering processes and technologies to make their data fit for purpose.

Table of Contents

Step 1

Think Big: Align Your Goals With The Organization

As with any major initiative, ensuring that your project is aligned with corporate goals has to be the first step. Because industrial data will be used by IT, operations, and line-of-business users, your objectives must support overarching business goals while being easily understood by data users across the organization. Cross-departmental collaboration will be necessary, as data architecture tends to reach across a great deal of the enterprise.

"Make sure the right cross-functional stakeholders are in the room from the project beginning."

IT, OT, and business users experience numerous benefits when they can more easily access and understand data. Their experience with industrial data has likely been characterized by lack of access and understanding, so guaranteeing access to contextualized, ready-to-use data will simplify and accelerate the work of IT and business users. Make sure the right cross-functional stakeholders are in the room from the project beginning, and that all stakeholders agree to prioritize the project and can reach consensus on the project goals.

Questions:

- How will end users benefit?
- How will the organization benefit?
- What is the best way to present those benefits?
- What is your timeline and budget?

Document your answers here:

Step 2

Get Strategic: Consider Your Architectural Approach

"Taking the time to plan your architecture before you start your project pays dividends as you expand to other areas and sites."

Lack of architectural strategy is one of the most common sources of failure for Industry 4.0 initiatives. A lot of well-meaning manufacturers start building integrations that perform their given tasks on day one, for the first project. But what happens when the custom code of the integration needs to be altered to receive new datasets? Or it needs to be duplicated or expanded beyond the pilot program? What about if a new data source or target is introduced? Any given integration can survive these challenges, but the time, effort, and cost of manually maintaining and scaling individual integrations always prohibits meaningful expansion eventually.

Taking the time to plan your architecture before you start your project pays dividends as you expand to other areas and sites. A well-planned data architecture will deliver the visibility and accessibility necessary to give your people and systems the data they need when they need it.

Questions:

- What are your data sources and target systems?
- How are your systems currently integrated?
- How will you scale your integrations?
- How will you know if data is not flowing or a connection is down?
- What does success look like, and how will you measure it?

Document your answers here:

Step 3

Start Small: Begin With A Use Case

Information Technology (IT) and Operations Technology (OT) projects should begin with clear use cases and business goals. For many manufacturing companies, projects may focus on machine maintenance, process improvements, or product analysis to improve quality or traceability. As part of the use case, company stakeholders should identify the project scope and applicable data that will be required.

In a factory, the use cases are typically driven by one of three key data structures: assets, processes, or products.

- Assets are the core data structure for use cases like predictive asset maintenance and tracking power consumption and emissions.
- Processes are the core data structure when looking at process control and monitoring line or cell production metrics.
- Products are the core data structure when looking at traceability, defect root cause analysis, and batch reporting.

"Identify the project scope and applicable data that will be required."

Each use case has a target persona who will consume this information and act on it. This persona is critical to identify because their knowledge, experience, and background will all impact how the data will be delivered to them—and ultimately the success of the project.

Questions:

- What will your first (or next) use case be?
- Who are the stakeholders and how will they benefit?
- What outcomes do you hope to produce with your use case?

Document your answers here:

Step 4

Identify The Target Systems

With the business goals and use cases identified, your next step is identifying the target applications that will be used to accomplish these goals. This approach is contrary to traditional data acquisition approaches that would have you begin with source systems. Focusing on your target systems first will allow you to identify exactly what data you need to send, how it must be sent, and at what frequency it should be sent, so you can determine which sources and structure are best suited to deliver that data. Focusing on the target system and persona will also help identify the context the data may require and the frequency of the data updates. Some target systems may require data from multiple sources blended into a single payload. Most require the data to be formatted in a specific way for consumption. By starting with the target system and persona, you can ensure that you are architecting your solution in a way that delivers not just the right data but the right data with the right context at the right time.

"Focusing on your target systems first will allow you to identify exactly what data you need to send, how it must be sent, and at what frequency it should be sent."

You can characterize the target application by asking these questions:

- Where is the target application located: at the Edge, on-premises, in a data center, in the Cloud, etc.?
- How can this application receive data: MQTT, OPC UA, REST, database load, etc.?
- What information is needed for this use case in this application?
- How frequently should the data be updated and what causes the update?

Document your answers here:

Step 5

Identify The Data Sources

Identifying the right data sources for your use case is a crucial step. However, there are often significant barriers to accessing the right data sources, including:

VOLUME

The typical modern industrial factory has hundreds to thousands of pieces of machinery and equipment constantly creating data. This data is generally aggregated within Programmable Logic Controllers (PLCs), machine controllers, or Distributed Control Systems (DCSs) within the automation layer, though newer approaches may also include smart sensors and smart actuators that feed data directly into the software layer.

"Gain a better understanding of the specific challenges you will need to overcome for your project by documenting your data sources."

CORRELATION

Automation data was primarily put in place to manage, optimize, and control the process. The data is correlated for process control, not for asset maintenance, product quality, or traceability purposes.

CONTEXT

Data structures on PLCs and machine controllers have minimal descriptive information—if any. In many cases, data points are referenced with cryptic data-point naming schemes or references to memory locations.

Furthermore, the context for telemetry data is often stored across transaction systems like MES, CMMS, QMS, and ERP. For example, the context for asset maintenance is typically in the CMMS, including the asset vendor, model, service date, and specification, while the context for a batch is in the MES including set points, alarms, product description, planned volume, and customer. This information is critical when taking the data outside the OT domain and delivering it to line of business users.

STANDARDIZATION

Automation in a factory evolves over time, with machinery and equipment sourced from a wide variety of hardware vendors. Hardware is typically programmed and defined by the respective vendor, resulting in unique data models created for each piece of machinery and a lack of standards across the site and enterprise.

You can better understand the specific challenges you will need to overcome for your project by documenting your data sources.

Characterize the data available to meet the target system's needs by asking these questions:

- What data is available?
- Where is the data located: PLCs, machine controllers, databases, etc.?
- Is it real-time data or informational data (metadata)?
- Is the data currently available in the right format or will it need to be derived?

Document your answers here:

Step 6

Select The Integration Architecture

Integration architectures fall in two camps: direct Application Programming Interface (API) connections (application-to-application) or integration hubs (DataOps solutions).

Direct API connections work well if you only have two applications that need to be integrated. The data does not need to be curated or prepared for the receiving application, and the source and target systems are static. Direct API connections are typically successful in environments in which the manufacturing company has a single SCADA or MES solution that houses all the information, and there is no need for additional applications to get access to the data.

"The DataOps approach to data integration and security aims to improve data quality and reduce time spent preparing and maintaining data for use throughout the enterprise."

Direct API connections do not work well when industrial data is needed in multiple applications like SCADA, MES, ERP, IIoT platforms, data warehouses and lakes, analytics, QMS, AMS, cyber threat monitoring systems, various custom databases, dashboards, or spreadsheet applications. Direct API connections also do not work well when there are many data transformations that must occur to prepare the data for the consuming system. These transformations can easily be performed in Python, C#, or any other programming language, but they are then "invisible" and hard to maintain.

Finally, direct API connections do not work well when data structures are frequently changing. When the factory equipment, programs running on the equipment, or target system requirements are frequently changing, direct API connections tend to fail and either stop working or leave holes in the data.

For example, a manufacturer may need to create short-run batches that require loading new programs on the PLCs. The products produced will require changes to the automation, the automation will likely need to be changed to improve efficiency, and in some cases, the equipment may need to be replaced due to age and performance.

Using the API approach buries the integrations, and therefore the ability to make changes to the automation, in code. Stakeholders may not even be aware of integrated systems until long after the equipment has been replaced or changes have been made, resulting in undetected bad or missing data for weeks or even months.

An alternative to direct API connections is a DataOps integration hub. The DataOps approach to data integration and security aims to improve data quality and reduce time spent preparing and maintaining data for use throughout the enterprise. An integration hub acts as an abstraction layer that uses APIs to connect to other applications while providing a management, documentation, and governance tool that connects data sources to all required applications.

An integration hub is purpose-built to move high volumes of data at high speeds with transformations being performed in real time while the data is in motion. Since a DataOps integration hub is an application itself, it provides a platform to identify impact when devices or applications are changed, perform data transformations, and provide visibility to these transformations.

As your use case scales, the integration hub methodology easily scales with it. With an integration hub, you can templatize connections, models, and flows, allowing you to more easily reuse integrations and onboard similar assets. Integrations hubs can also share configurations, projects, and more, so when you are ready to expand your use case to another site, you can simply launch another integration hub and carry your work over.

Selecting Your Integration Architecture

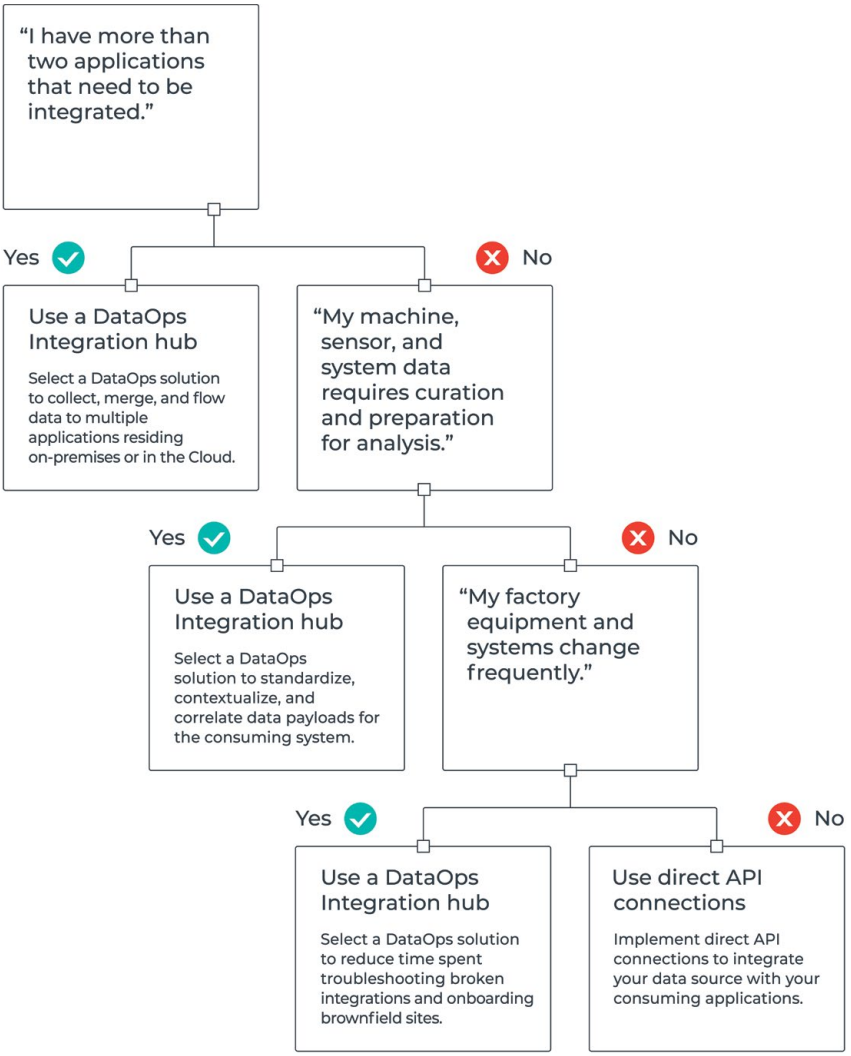


Figure 1: Integration hubs are an alternative to using Application Programming Interfaces (APIs) solely when managing the many types of data associated today with automation projects.

Questions:

- Where does your current integration architecture excel?
- Are there areas in your organization held back by your current integration architecture?
- On a scale of 1-10, how would you rank your ability to quickly create or adapt integrations? Why?

Document your answers here:

Step 7

Establish Secure Connections

Once your project plan is in place, you can begin system integration in earnest by establishing secure connections to the source and target systems. It is vital that you wholly understand the protocols you will be working with and the security risks and benefits that come with them.

"Wholly understand the protocols you will be working with and the security risks and benefits that come with them."

Many systems support open and secure protocols to define the connection and communication. Typical open protocols include OPC UA, MQTT, REST, ODBC and AMQP—among others. There also are many closed protocols and vendor-defined APIs for which the application vendor publishes the API protocol documentation. Some protocols and systems support certificates exchanged by the applications. Other protocols support usernames and passwords or tokens manually entered into the connecting system or through third-party validation. In addition to user authentication, some protocols support encrypted data packets, so if there is a "man in the middle" attack they cannot read the data being passed. Finally, some protocols support data authentication. Then, even if the data is viewed by a third party, it cannot be changed.

Security is not just about usernames, passwords, encryption, and authentication. It is also about integration architecture. In many companies the controls network is separated from the business network by at least one firewall (if not two) and a DMZ (the space between the two networks). Some companies will have even more networks to separate traffic and secure systems. Protocols like MQTT require only outbound openings in firewalls, which security teams prefer because hackers are unable to exploit the protocol to get on internal networks.

Questions:

- Does your communication protocol support secure connections, and how are these connections created?
- Where does your data move to and from, and how exposed is it during movement?

Document your answers here:

Step 8

Model The Data

The corporate-wide deployment and adoption of advanced analytics or other Industry 4.0 use cases is often delayed by the variability of data coming off the factory floor. As we noted in Step 5, each industrial device may have its own data model. Historically, vendors, systems integrators, and in-house controls engineers have not focused on creating data standards; they have refined the systems and changed the data models piecemeal over time to suit their needs. This approach worked for one-off projects, but today's IIoT projects require more scalability.

"Define the standard data set required in the target system to meet the project's business goals."

The first step in modeling data is to define the standard data set required in the target system to meet the project's business goals. The real-time data coming off the machinery and automation equipment is at the core of the model. Most of the real-time data points will map to single-source data points, but when a specific data point does not exist, data points can be derived by executing expressions or logic using other data points. Data also can be parsed or extracted from other data fields, or additional sensors can be added to provide required data.

These models also should include attributes for any descriptive data, which are not typically stored in the industrial devices but are especially useful when matching and evaluating data in the target systems. Descriptive data could be the machine's location and asset number, unit of measure, operating ranges, asset vendor, last service date, or other contextual information.

Once the standard models are created, they should be instantiated for each asset, process, and/or product. This is typically a manual task, but it can be accelerated if the mapping already exists in Excel or other formats, if there is consistency from device to device that can be copied, or when parametric definitions can be applied.

Some source data must be pre-processed or conditioned prior to being mapped into an instance. The data may not be readable in its current state and may require a transformation or third-party library to convert it. Or, perhaps the data must be aggregated across a time period or based on an event occurrence, and then undergo calculations performed like average, max, or count.

Questions:

- What descriptive data do your target systems need?
- How will you implement your models once they're created?
- What additional processing of the raw data is required to make it usable in a model?

Document your answers here:

Step 9

Flow The Data

When the instances are complete, data flows control the timing of when the values for an instance are sourced, standardized, contextualized, calculated, and sent to the target system. This is typically performed by identifying the instance to be moved, the target system, and the frequency or trigger for the movement. Three publishing examples are provided below.

CYCLIC PUBLISHING

Some systems require complete data sets published to them at a consistent frequency, which might range from milliseconds to days. Cyclic publishing is used when publishing to systems, such as databases, data warehouses or data lakes, that use time-based trends and analysis to visualize the data.

EVENT PUBLISHING

In event publishing, a complete set of information is assembled and published when a given event occurs, notifying necessary systems and/or logging the event. Events might range from notifications that work cell has completed a part or the temperature reading exceeds set limits.

TIME SERIES PUBLISHING

Time series publishing only publishes when changes occur. It is often the lightest load on the network and storage, but lack of context can make reconstructing information for analytics or visualization challenging. Many Operational Technology (OT) systems and some IT systems can consume and reconstruct time series data but require domain knowledge to do this.

In addition to collecting, modeling, and flowing data, some system integrations require more complex processing of the data. This can include buffering data, compressing it, and sending it in a file to a data lake. It also may include complex sequencing of multi-step processes. These activities are best performed through a graphical staged pipeline builder that allows for multiple reads and writes in a single pipeline, has many predefined steps as well as core data transformation steps, and can store pipeline state across runs.

"A well-built data flow will retain the semantics of a model while transforming the presentation and delivery to the unique needs of the systems consuming it."

A well-built data flow will retain the semantics of a model while transforming the presentation and delivery to the unique needs of the systems consuming it. Over time, data flows also will require monitoring and management. Managing your flows in an integration hub allows you easily track changes across iterations and scale changes via templating.

Questions:

- Which flow triggers will you use and why?
- How will your flows alter your model's presentation for consuming systems?
- What additional processing is required to optimize or sequence interactions to enable the required integrations?
- How will the data be organized in the target system and how can this be identified in the payload?

Document your answers here:

Step 10

Chart Your Progress And Expand To New Areas

Now that your data is flowing and your use case is officially off the ground, it is vital that you pay close attention to your results and track your progress. It might seem like an obvious step, but gathering performance data may not always be as simple as it seems. New Industry 4.0 use cases often explore entirely new functionality, so processes to track your progress may not presently exist in your organization.

"New Industry 4.0 use cases often explore entirely new functionality, so processes to track your progress may not presently exist in your organization."

Questions:

- How will you chart your progress?
- Can your flows be easily scaled across the site or enterprise? Why or why not?
- How will you present ROI to your peers?

Document your answers here:

Wrap Up

Factories and other industrial environments change over time. Equipment is replaced, programs are changed, products are redesigned, systems are upgraded, and new users need new information to perform their jobs. Amid this change, OT and IT professionals will collaborate on new projects aimed at improving factory floor productivity, efficiency, and safety. You will need industrial data that is fit for purpose to make the data useable, and you will need tools to accomplish this task at scale—like a DataOps integration hub. By using an integration hub, administrators can evaluate equipment and system changes and identify integrations that must be modified or replaced. They can make changes to data models and enable new flows in real time.

Making industrial data fit for purpose will be critical to manufacturers interested in scaling their Industry 4.0 projects, handling data governance and ultimately driving digital transformation. I hope this workbook serves as a practical guide to getting started on your next project and designing an agile architecture. Remember: Think big, start small, and scale fast!

About HighByte

HighByte is an industrial software company founded in 2018 with headquarters in Portland, Maine USA. The company builds solutions that address the data architecture and integration challenges created by Industry 4.0. HighByte Intelligence Hub, the company's award-winning Industrial DataOps software, provides modeled, ready-to-use data to the Cloud using a codeless interface to speed integration time and accelerate analytics. The Intelligence Hub has been deployed in more than a dozen countries by some of the world's most innovative companies spanning a wide range of vertical markets, including food and beverage, health sciences, pulp and paper, industrial products, consumer goods, and energy.

Learn more at https://highbyte.com.

HighByte

Novotek