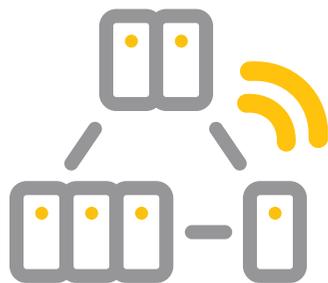


# Why IIoT Projects Fail

## 3 Secrets to solving the chokepoints



# INTRODUCTION

Is IIoT all it's cracked up to be?

When you ask people what they're doing with the Industrial Internet of Things, you rarely hear that a company has put it into action with much success. If you've tried, or are thinking about trying to do an IIoT project, here's what you need to know to get your project up and going in DAYS versus weeks or months.

In this white paper, we'll explore the three fundamental reasons your IIoT projects fail to see the light of implementation, and then reveal the three secrets to overcome each of these hurdles.

**Connecting devices is easier said than done.** Even if devices are connected, there are often no simple tools to manage the device and get data out of one 'language' and into another. For example, how do you cleanly and easily get your programmable logic (PLC) data into your enterprise resource planning (ERP) system?

**Custom code puts shackles on your automated system.** Getting all of your legacy devices talking to all of your modern devices, and then connecting that network to your automated system takes a lot of custom code because you must account for all the different protocols and brands.

**IIoT by itself doesn't give you scalability.** Scalability becomes expensive when you have to add more computers, servers or nodes to handle more and more devices as your deployment grows.

Projects stall, or even fail, as your team wades through these issues. Manufacturers that learn the secrets to overcoming these three chokepoints will be the ones that win at IIoT. This white paper will show you how to get there, and how to get there quickly.

## Contents

---

<b>Introduction</b>	2
<b>Chokepoint #1</b>	3
Getting Things Connected is Easier Said than Done	3
You Have to Embrace Complexity	3
You Have to Deal with Latency	4
Data Isn't Always Accurate	4
You Have to Make Legacy Devices Talk with Modern Equipment	4
<b>Chokepoint #2</b>	5
Custom Code for Interoperability Puts Shackles on Your Ability to Move Data	5
Time Kills Projects	5
Subject Matter Experts Can't Operate with Agility	5
You're at Risk When Only a Handful of People know the Code	5
<b>Chokepoint #3</b>	6
IIoT Itself Doesn't Give You Scalability	6
You Have to Navigate Hardware Limitations and Software Costs	6
Making Changes Gets Riskier to an Ever-Growing System	6
<b>Secret #1</b>	7
Use a Data-Centric IIoT Software to Map Devices Directly to Applications (for Other Devices)	7
Native Drivers That Go Beyond APIs, OPC, and MQTT	7
Native Drivers	8
Make Sure You Get Drivers for Apps, Too	8
Instantly Get Legacy Devices Talking with Modern Devices	8
<b>Secret #2</b>	8
Use a Visual Flow Chart to Make Data Accessible to All Users	8
Accessible to Everyone, No Programmer Required	9
Make Sure Tags Can Be Auto-Enumerated	9
All Users Can Be Agile with IIoT	10
<b>Secret #3</b>	10
Cut Down the Number of Devices and Software Instances to Achieve Scale	10
Supercharge Your Edge Devices for Global Scale-Out	10
Scale Beyond the Factory Floor	11
<b>Summary</b>	11

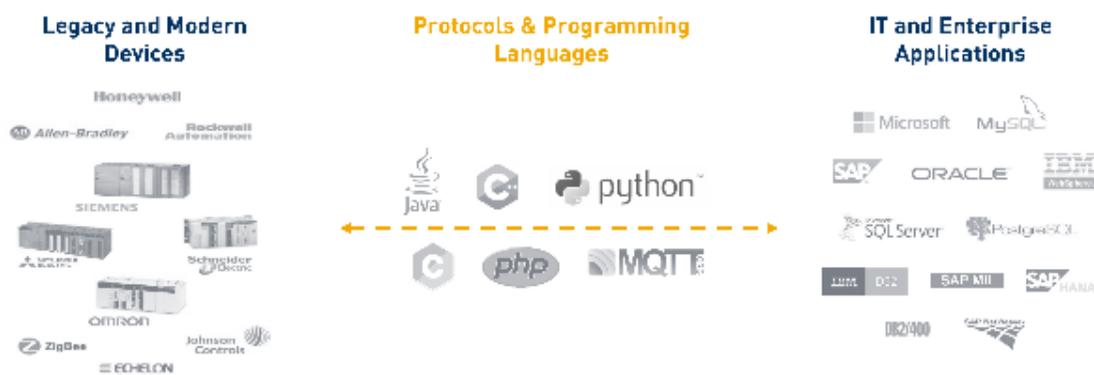
# Checkpoint #1

## Getting Things Connected is Easier Said than Done

Why is it so hard to get connected? Many IIoT vendors gloss over this. They say, “get connected,” and then tout all the great analytics and services you can do. Once manufacturers take the bait and decide to take the plunge, they suddenly realize that connecting to all these different devices – legacy, modern, proprietary and open source – is really difficult! Not impossible, but the tedious set-up has blocked many companies from getting IIoT up and running.

If you’ve ever engineered systems on the plant floor, you know there are those “things” down on the plant floor that are a nightmare to connect to and integrate with a variety of other applications. A simple data collection task can take weeks of coding.

With an exploding demand for real-time manufacturing data, a huge backlog of data collection tasks sits in a wish-list queue. Data quickly needs to be piped when, where and how you need it, so that someone can actually do something with the data.



A simple data collection task can take weeks of coding to make modern, legacy, proprietary and open source devices and applications all work together.

## You Have to Embrace Complexity

Over time, the industrial plant floor has evolved as technology has evolved. This brought dramatic improvement to automated processes. For better or for worse, this advancement also means more complexity. This complexity is not going away; in fact, it will increase!

Complexity by definition is not a bad thing. Complexity is generally used to characterize something with many parts which interact with each other in multiple ways, culminating in a solution greater than the sum of its parts. Embracing complexity means we accept there are a lot of moving parts in IIoT solutions that we need to link together for success.

As a result, plant floors have a mixture of device brands and different protocols. Each mixture is totally unique to each company. Many devices also use proprietary data sets, making it very difficult to get different brand PLCs to share data. Some machines were never built to be connected, like CNC machines.

There is no one standard way of connecting everything together. In response to this industry need, OPC was introduced. But as we'll see next, it too brings a host of challenges in data speed and accuracy.

### **You Have to Deal with Latency**

OPC was designed to provide industrial automation with a standard networking protocol that requires polling to receive data from devices. Polling is where the system must "ask" the device for data over and over again at a preset rate, such as once every second or once every half hour.

OPC also requires multiple steps to send the data. It goes from a device, to a PLC, to an OPC server, which then sends it to an OPC client. Then, the OPC client sends it to the on-premise server or cloud network to be utilized and processed. Additionally, IT support is required for the servers, computers and software running the OPC server and OPC client software. Additionally, PLCs must have separate connections to every other thing or software application. So connections must be written from PLC 1 to thing 1, PLC 1 to thing 2, PLC 1 to thing 3, and then repeated for PLC 2 and its connection to each thing. If you want to get PLC data into an enterprise resource planning (ERP) software, then you'd have to write a connection from PLC 1 to ERP software 1, ERP software 2, and so on.

Add in polling to all those layers, and it's a recipe for tremendous latency.

### **Data Isn't Always Accurate**

Another challenge is data accuracy. OPC offers no feature to guarantee data accuracy. One company with a pharmaceutical plant in Jacksonville, Florida, was up against this challenge. They were using OPC to poll devices, and often received 3,000 data packets for each production run. They had no way to verify which packet was the correct packet to match the finished product lot. To answer this, the engineering team wrote a lot of complex, custom code to run double-checks on the data sources and the receiving ends to ensure the data matched. It's a lot of work to create and maintain.

Another challenge is data accuracy. OPC offers no feature to guarantee data accuracy. One company with a pharmaceutical plant in Jacksonville, Florida, was up against this challenge. They were using OPC to poll devices, and often received 3,000 data packets for each production run. They had no way to verify which packet was the correct packet to match the finished product lot. To answer this, the engineering team wrote a lot of complex, custom code to run double-checks on the data sources and the receiving ends to ensure the data matched. It's a lot of work to create and maintain.

### **You Have to Make Legacy Devices Talk with Modern Equipment**

MQTT is quickly becoming one of the top protocols for IIoT, and modern devices are coming built with MQTT. However, the only way to take advantage of MQTT is to buy MQTT-enabled devices. No one wants to rip and replace legacy devices they've already amortized for 20 or 30 years just to get all new MQTT devices.

When you want to add new equipment at some point, such as the latest sensor that's the hottest item on the market, it will most likely be designed with MQTT in mind. The downside is that you're going to have to do extra work to make MQTT devices work with your legacy devices. To the thousands of devices you might have in a plant, you may have 10 devices equipped with MQTT. Migration of all devices to MQTT will be slow, and doesn't solve today's problem of connecting legacy and modern devices.

Either way, you're going to need to do more coding and more coding to make more and more new things work with devices that are just getting older and older. But you'll see later on in this white paper how to bridge this gap.

## Chokepoint #2

### **Custom Code for Interoperability Puts Shackles on Your Ability to Move Data**

The complex nature of a variety of vendors and protocols, even while using OPC and MQTT, still requires the use of custom code. Since custom code is necessary, this becomes the second chokepoint that dries up IIoT initiatives.

### **Time Kills Projects**

Projects often get backlogged in priority because they simply take much too long to code. When the length of time is weighed against the value of the project, it can get killed. This happens more than one might think.

There's also the reality that there are other higher-priority projects already in the cue, also suffering from a long wait in line as programmers tirelessly code away.

### **Subject Matter Experts Can't Operate with Agility**

Often, the need for a programmer is the stop-gap between a subject matter expert and getting an improved process or product. Their job is to analyze the process and determine necessary changes. If he knows what needs to change but can't get it implemented into the process, it's a lost opportunity for better or more product coming off the production lines.

For example, in a dairy plant that's producing cheese, data such as temperature of the cheese, temperature in the facility, product density, etc. may be tracked. The product specialist views the data and determines that the process needs to be tweaked; maybe the speed of a machine needs to be slowed or sped up. He doesn't want to worry about going to the programmer, who says he's too busy, and put a ticket it in and wait a few weeks.

### **You're at Risk When Only a Handful of People know the Code**

When custom code is the foundation for your plant floor operations, it poses a couple risks. The first is that it's difficult to make changes to a live system without causing downtime. The second is that one or a few special people have now written the code, and only they are familiar with how everything is coded – when they are unavailable or leave the company, the effects can be detrimental.

Another reality is some devices run great for two or more years and engineers don't have a reason to work with them. With so much time in between working with the device, engineers forget how to work with that specific device, and have to spend time re-learning all the do's and don'ts of getting the device to work.

As more and more functionality is added over time, you'll end up with a monster application that only a select few know how change programmatically. Your system will never be agile or flexible or any of the things you really want it to be.

## Checkpoint #3

### IloT Itself Doesn't Give You Scalability

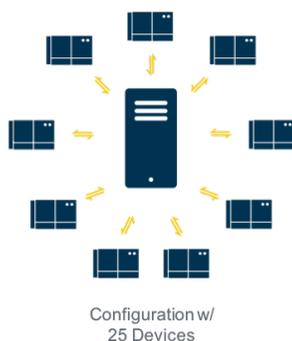
One of the prevailing thoughts behind IloT is a seamless, complex network that reaches to the far depths of your enterprise: from all the way down to a few vibration sensors on a conveyor belt to as far outside the plant floor as weather data from drones flying over your supplier's fields. Everything can and should be connected, in theory.

But IloT in general doesn't give you scalability. Expanding your connected network is a double-edged sword. You want to make it wider and deeper, but at the same time, it just doesn't scale that way.

### You Have to Navigate Hardware Limitations and Software Costs

With most industrial automation software or IloT software, users have to buy multiple software instances. This is because one node or server can only handle a certain number of devices. In order to achieve scalability, multiple nodes or servers need to be purchased, each with an instance of the software running on it.

Costs for every server, every node, licenses for each software instance, the cost for IT to maintain those items, etc. start to really add up. Scalability becomes very expensive when you have to architect your system to handle more and more devices as your deployment grows.



One configuration node or server can only handle a certain number of devices, making it expensive to scale an IloT deployment.

### Making Changes Gets Riskier to an Ever-Growing System

Most changes and updates to an automated system require the engineer to touch the live system. This means that changes need to be scheduled or done during planned downtime in order to add it. Then hope everything works fine when the system needs to go back up. This is very disruptive.

Updates can't be done on the fly, and often the existing framework is upset and engineers are dealing with unknowns under a high-pressure setting to get a high-value production line back up. The bigger the system, the more difficult it is to update and upgrade because there's so much more at stake.

Now that we've covered the three biggest chokepoints to IIoT, let's look at how you can skip right past these with a data-centric IIoT software platform.

## Secret #1

### **Use a Data-Centric IIoT Software to Map Devices Directly to Applications (or Other Devices)**

This seems pretty straight-forward, but as you've probably realized from the three chokepoints we discussed above, you simply cannot map devices directly. Things just don't plug-and-play like you'd expect.

So how do you map devices, like PLCs, directly to applications? You need a data-centric IIoT software platform that is designed to do just that. A data-centric platform means it was designed from the ground up to merge legacy and modern devices regardless of communication protocol, and provide a central data pipeline to all devices and applications on the network, giving you complete control over how, when, and where your data is used.

Many IIoT platforms focus on analytics, but they are data-poor due to their inability to quickly on-board devices. You might call these "analytic-focused" IIoT platforms. But without great data, users have great tools they are continually frustrated with because they can't get real answers out of inaccurate or incomplete data.

A data-centric software platform will let you truly get the data you need, and pipe it into any analytical application you already own, or would like to purchase in the future.

### **Native Drivers That Go Beyond APIs, OPC, and MQTT**

Every IIoT platform will have standard tools for APIs, OPC and MQTT, but they often don't have many native drivers. A data-centric platform will have a broad array of native drivers. The keyword here is "native". This means that there are drivers already written, so you don't have to. The lack of native drivers is what leads programmers to have to write all that custom code. If you choose a software system that has all the drivers you need, getting your devices on-boarded will finally feel like what you always wanted it to be: plug-and-play perfection!

Remember: Don't be lured into the idea that APIs and standard protocols will give you flexibility. They never get rid of custom programming. APIs, OPC and MQTT are an advertisement that you'll need to do custom programming. Native drivers mean that all the heavy lifting of custom programming is already done for you.

Now you can plug devices into your platform and they'll instantly play well with any other device or application. This is what will allow you to get your IIoT projects done in a matter of days, instead of months.

APIs, OPC & MQTT	Native Drivers
Require custom programming	No programming required
Polling for legacy devices using OPC	No polling
Latency due to polling	No latency

### Make Sure You Get Drivers for Apps, Too

One short note: when looking for an IIoT platform, also be mindful that it includes native drivers for analytical software platforms, like SAP, Microsoft Azure, etc. This will prevent you from having to write custom code on the application end.

### Instantly Get Legacy Devices Talking with Modern Devices

Pre-made, native drivers all on one IIoT software platform means that all your devices and applications will just talk. Anytime, anywhere, and just as easily as it is to add a new app to your mobile phone and make it work with everything else on your phone. Isn't this what you thought IIoT is supposed to do?

Without having to depend on solely APIs, OPC or MQTT for device connectivity, you won't have to deal with polling, system latency caused by polling, or custom code. You'll have a cleaner, faster network that zips data around at lightning speed. PLCs can talk to other PLCs. PLCs can talk directly to ERP software applications. And vice versa! You'll also be able to instantly add in new devices that speak MQTT, without worrying about how to make them talk to your legacy devices.

This means you can spend your time on labor that gets your system doing something meaningful, instead of wasting time hassling with just getting devices to talk with each other.

## Secret #2

### Use a Visual Flow Chart to Make Data Accessible to All Users

Once devices are on-boarded easily, the next hurdle you need to overcome is system configuration. This can take a lot of custom coding, even if you have native drivers for every device and application.

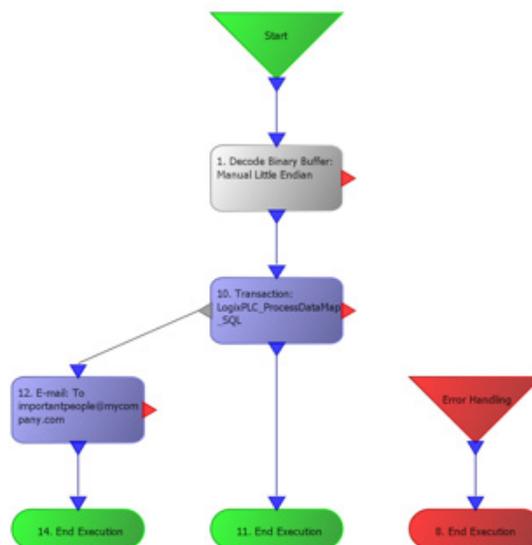
Not only should a data-centric IIoT software platform make data accessible to every device and every application on the network; it should also make the data accessible to you, the user. You should be able to control your data and the details of when, how and where it will be used; all without being a programmer.

## Accessible to Everyone, No Programmer Required

By using an easy drag-and-drop visual flow chart, an IIoT platform can make it easy to build any kind of logic without custom programming. This means that your plant engineers and even subject matter experts can have the ability to fully customize a deployment without the need to get a programmer to make the change.

Since this creates easy access for any user, you'll also want to have a method ensuring only the right people make the right changes, and the wrong ones don't. The software should also allow you to set permissions on a role basis, so certain parts of your process can only be changed by certain people. This keeps the system secure from just anyone doing anything to your system.

A robust software platform should also give you advanced features to get behind the scenes to customize things beyond the visual flow chart, but for rare occasions when you want nitty gritty control. You'll need a programmer at this level, but now you're expanding your system's features with his time, not just asking for normal run-of-the-mill system logic changes.



Using a visual flow chart means you don't need to rely on a programmer to make business logic changes to your IIoT system.

## Make Sure Tags Can Be Auto-Enumerated

A data-centric IIoT software platform can make tags appear instantly upon device connection, and format the tags into a universal naming architecture via a feature called "tag auto-enumeration".

The software makes process variables, like temperature or pressure, look the same regardless of the type of PLC. In the event of a legacy PLC with linear addressing – for example, Siemens S7-300, DB3[0] – the tag auto-enumeration feature provides a simple mechanism for aliasing this information into a more human-friendly naming structure.

This makes it easy for users to take a temperature tag from any device, and tell the system what to do with that tag. No stress is caused over mismatched names of tags that may cause a disruption in the automated system logic.

## **All Users Can Be Agile with IIoT**

A data-centric IIoT software platform focuses on making data run smoothly in parallel to your live system, as opposed to being embedded in the custom code that is running the live plant itself. This fundamental separation of the digital plant floor makes your entire team of users much more effective and agile

When data is in a safe “pipeline” that is decoupled from your running system, changes can be made on the fly. You never need to touch the live system. You’ll never need to schedule downtime to make cutovers to new code, or cause unexpected downtime because something in the code broke the system. You can simply and easily make adjustments without causing disruption to any part of the system.

## **Secret #3**

### **Cut Down the Number of Devices and Software Instances to Achieve Scale**

With the more devices you can now easily connect, and the more data you can more easily direct the flows, your last hurdle is how to handle the increased data load on your hardware.

A data-centric IIoT software platform solves this. It takes the load off the hardware, because it handles the data dramatically more efficiently. As a result, you save tremendously on the number of computers you need to run the same IIoT deployment. Think about the cost you’ll save if you go from 1,000 computers to 1, maybe 2 for a redundant system. The cost for IT to administer those computers is gone. The cost for electricity to run those computers each year is gone. And the cost of licensing the software for each instance is gone.

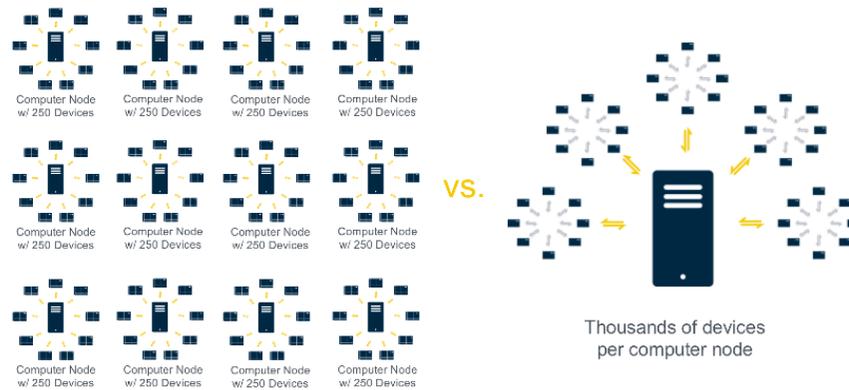
Now, you can scale without the cost getting in your way.

### **Supercharge Your Edge Devices for Global Scale-Out**

Edge device logic is where things get really interesting with a data-centric IIoT software platform. Since you now have direct drivers to every device, you no longer have to use OPC. Instead of using OPC to poll data and continually “ask” your devices to send data every millisecond, your devices can now make decisions on when to send data according to pre-set configurations. Only one data transaction occurs, instead of hundreds back and forth. This guarantees data accuracy, and can practically eliminate system latency.

You can even set up one PLC to do something based on another PLC’s variables, because they can talk directly to each other. You no longer have to bounce one PLC’s data to the central server, wait for it to get processed, and then wait for the central server to tell another PLC to do something. This may take milliseconds, and might not seem like a lot, but when those milliseconds add up across hundreds or

even thousands of devices, it causes a lot of network congestion. So now, think about how much faster, and more accurate, your system will run without the central server having to process all that data back and forth from devices to server back to devices.



Setting up edge logic on your devices greatly reduces the load on your hardware.

## Scale Beyond the Factory Floor

Once you empower scalability by leveraging your device data on the edge more efficiently, you can scale IIoT beyond the factory floor. Think what you could do outside the “four walls” of your plant.

Your manufacturing process is a pivotal point along your supply chain. Connecting your automated process to other points along the supply chain is where the future of IIoT is heading. Enterprises are making leaps and bounds in innovation by building out IIoT deployments across every point of their supply chains.

You can extend your IIoT solution into everything before, and after, your manufacturing process. Think about the materials your plant receives and how you can improve when and how it’s delivered from your suppliers. Think about how you could differentiate your business by offering customers a new service with your product. What about communicating production inventory with your sales department, so they know what’s available to sell, what’s backlogged, and what might take too long to fulfill? The possibilities for innovation are endless.

## Summary

When you use a data-centric IIoT software platform to implement the secrets we explored above, you’ll never have another IIoT project that fails due to technology restraints. We’ve included a checklist to the right that will help you evaluate IIoT software platforms against the points we’ve reviewed in this white paper.

# Checklist for Choosing an IIoT Software Platform

## On-Boarding Devices

- Does it have a library of all the drivers you need?
- Beyond APIs, OPC or MQTT. Does it have device AND application drivers?
- Can it make any PLC talk to another PLC, directly?
- Can data move bidirectionally?

## System Configuration

- Does it have a visual flow chart so you dont have to write code for logic?
- Do tags automatically populate the system upon device connection?
- Do tags get formatted in a universal naming format?
- Can you make changes on the fly, without touching the rest of the live system?

## Scalability

- Can it enable devices to make condition-based decisions on when and where to send data?
- How much data can the software handle per software instance? Can you do everything on just a few servers?
- Can it be installed directly on a PLC, in addition to a server, rack or any other edge device?

# deviceWISE<sup>®</sup>

## FOR FACTORY

# No Custom Coding. Period.

**Build a fully robust IIoT solution without writing a single line of custom code. Now everyone in your organization can quickly advance with IIoT.**

Telit's deviceWISE for Factory is a data-centric IIoT software platform that ensures your deployment starts with the right data, and empowers you to decide how, when and where you want to use it.

Your devices will become "wise", so they know exactly how to talk with any other device or application instantly via deviceWISE's data-mapping pipeline – it's almost magical. With the data pipeline in place, anyone can easily build advanced business logic without writing a single line of custom code.

- ✓ **Make Devices Talk Directly**  
Any device can be connected in minutes with the largest library of native drivers.
- ✓ **Accurate Data Every Single Time**  
No polling, no latency, and get condition-based data messaging to ensure complete data accuracy.
- ✓ **Complexity Solved**
- ✓ **Drag-and-Drop Development**  
Quickly create visual flow charts to automate every aspect of your IIoT solution.
- ✓ **Flexible Enough to Handle Any Size Scale-Out**  
Make every device work smarter so your system doesn't get held back by central-server limitations.

[www.telit.com/no-more-coding](http://www.telit.com/no-more-coding)