



Securing the Purdue Model

How to close the gaps in your OT data strategy



Cogent
DataHub™

The Purdue Model has been modified to meet modern security requirements of segmenting OT and IT networks by using DMZs, data diodes, or other means. However, standard protocols like MQTT and OPC UA cannot maintain a consistent data set across a DMZ or data diode, which can lead to incorrect AI outputs and costly errors in business decisions. A tunnel/mirror approach provides full support for DMZs and data diodes, while creating no attack surfaces and guaranteeing data consistency across the entire Purdue Model.

Key takeaways

- 🎯 No IT system should have a direct network path into the OT zone. Revising the Purdue Model by introducing Level 3.5 allows for a DMZ and/or data diodes to isolate OT and IT networks.
- 🎯 The two most common OT protocols (OPC and MQTT) are inadequate for DMZ and data diode architectures. Whether used alone or together, they cannot guarantee consistent data from OT to IT, leading to costly mistakes arising from incorrect or stale production data.
- 🎯 Cogent DataHub software ensures data consistency over secure data connections between each level of the Purdue Model. When the data is consistent system-wide, AI analysis and forecasts stay consistent and correct.

Levels of the Purdue Model

The Purdue Model (or PERA, the Purdue Enterprise Reference Architecture) was developed in 1992 and has been used ever since to organize the logic and flow of data between the various elements of industrial process control and enterprise systems. It divides these elements into a hierarchy of levels according to function from the physical process to business planning.

Originally there were five levels: the physical process itself, three levels of control (low-level, supervisory, and site-wide), and business planning. With the advent of IoT, Industry 4.0, and cloud services, it soon became clear that a new level was needed for a DMZ (De-Militarized Zone), and for the enterprise and cloud.

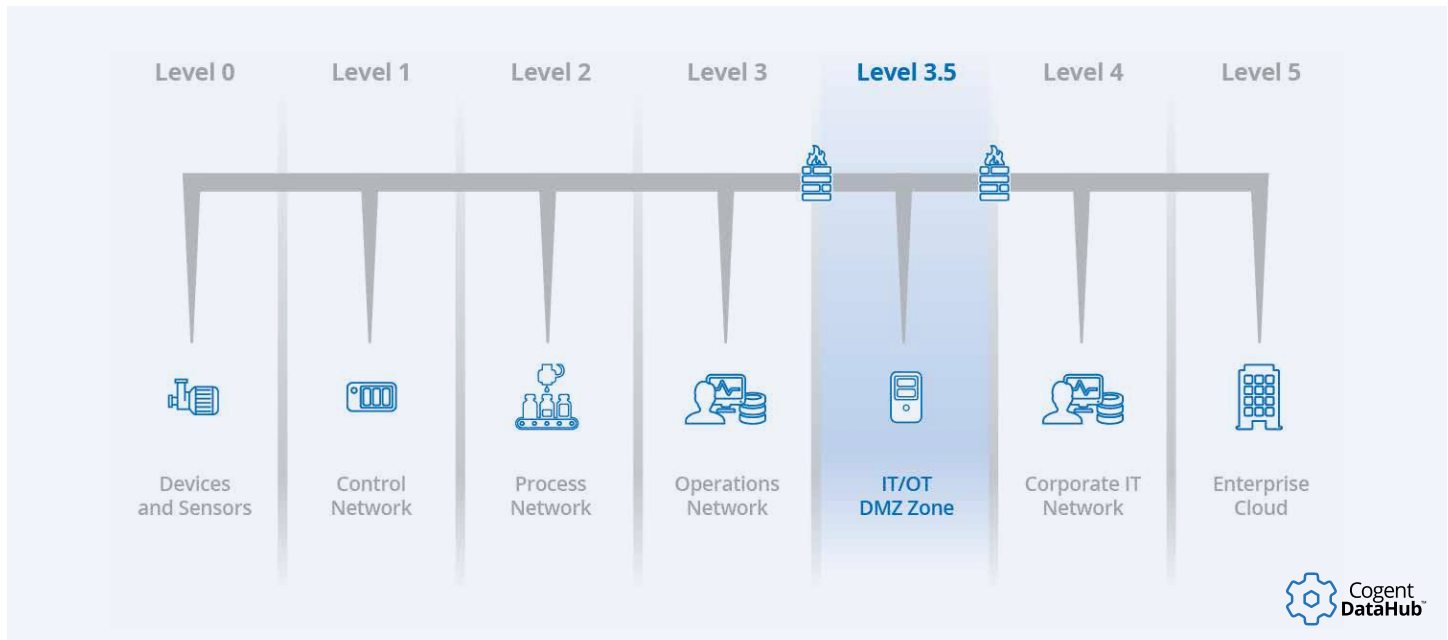
New DMZ level: What does it mean for you?

Adding a DMZ level creates a critical security boundary in the Purdue Model. Its purpose is simple: ensure that no system in the IT levels has a direct network path into the OT levels. All traffic crossing the IT/OT boundary must terminate in the DMZ first. Otherwise, the OT network is directly exposed to any security breach in the IT network, such as phishing emails or ransomware attacks, which can cost millions of dollars to recover from.

Using a DMZ is vital to maintaining the highest IEC 62443 security levels 3 and 4 for SCADA systems and sensitive instruments. The NIS 2 Directive and NIST CSF 2.0 recommend isolating networks, which is most effectively done using a DMZ. NIST SP-800-82 says, “The most secure, manageable, and scalable control network and corporate network segregation architectures are typically based on a system with at least three zones, incorporating one or more DMZs.”

The January 2026 CISA/NCSC-UK joint guidance goes further, explicitly stating that all connections with the OT environment should be initiated as outbound connections from within the OT environment. A DMZ enforces this by acting as the only permitted transit point, greatly reducing the attack surface of both OT and IT systems.

Based on these recommendations, today's Purdue Model looks like this:



0 - Physical process: Devices, machinery, sensors that do the work.

1 - Low-level control: PLCs, RTUs, translate commands to the machines.

2 - Supervisory control: SCADA systems, DCS, HMIs enable operator interaction.

3 - Site-wide control: MES systems, plant historians integrate plant data.

3.5 - DMZ: Tunnel/mirror, data diode, and pub/sub technologies secure data between levels.

4 - Business planning: ERP systems, business logic, execute planning and direction.

5 - Enterprise: corporate IT, WAN, and email connect to the Internet and outside world.

Where Cogent DataHub operates

Cogent DataHub tunnel/mirroring capabilities make it ideal for deployment in the DMZ, where it can act as the controlled egress point for OT data crossing the IT/OT boundary. But that is only part of the picture. Cogent DataHub software

also operates at Levels 1, 2, 3, 4, and 5, performing functions that uphold the integrity of the Purdue architecture at each level.

At Level 1 and 2, a DataHub instance connects to sensors, PLCs, RTUs, SCADA systems, DCSs, and HMIs through OPC DA, OPC UA, MQTT, and Modbus. It collects real-time data from supervisory systems and makes it available within a unified namespace at the local level. This means that even at the low-level and supervisory control layers, data from multiple protocols can be integrated, consolidated, and stored without requiring each application to support every protocol natively.

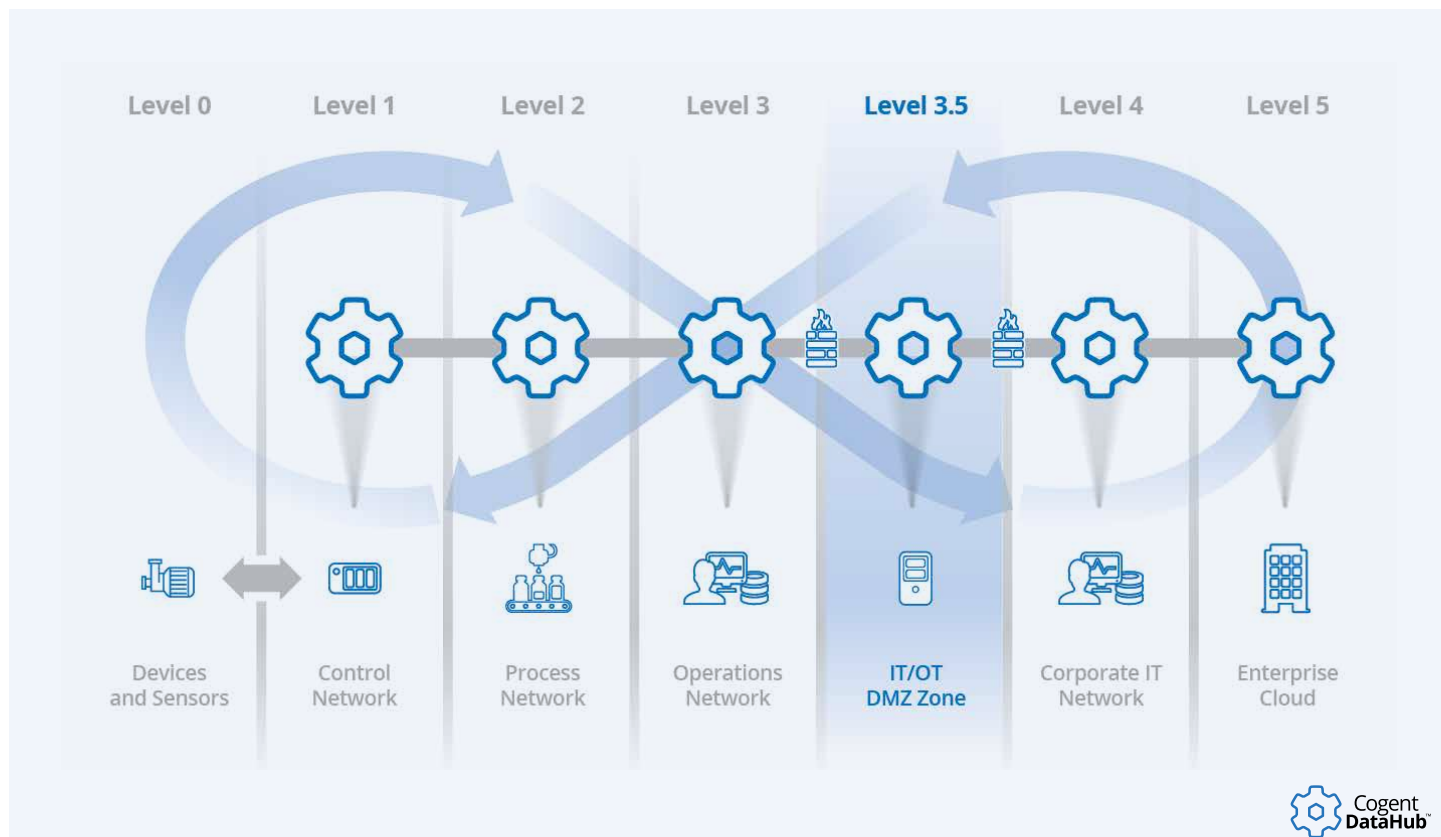
At Level 3, DataHub software performs site-wide data integration. It aggregates data from multiple Level 2 sources — OPC servers, SCADA historians, Modbus devices — into a single namespace that MES systems, plant historians, and reporting tools can consume. Importantly, a DataHub instance at Level 3 initiates outbound tunnel/mirror connections to the DMZ at Level 3.5. No inbound firewall ports need to be opened on the Level 3 network.

At Level 3.5, a DMZ-hosted DataHub instance receives these outbound connections and mirrors the full data set.

It then serves data to Level 4 and Level 5 consumers — ERP systems, analytics platforms, cloud services — without any of those systems having a direct network path into the OT zone.

At Level 4 and 5, another DataHub instance can receive data from the DMZ and distribute it to business systems, making OPC, MQTT, and historical data available to enterprise applications that would otherwise have no way to access plant-floor information.

The result is a chain of DataHub instances that spans the Purdue hierarchy while respecting every boundary. Each instance mirrors the full data set at its level, maintains guaranteed data consistency, and provides local data access to qualified clients. If any link in the chain breaks, every downstream data user is notified immediately. When the connection is restored, data quality returns to Good across every node—automatically.



Outbound connections: No inbound ports opened

Making outbound connections from the OT network is the most secure way to access production data remotely. Any open incoming firewall port constitutes a security exposure.

Most industrial protocols were not designed for this. OPC UA, for example, is based on a client-server model where the client initiates an inbound connection to request data from a server. That server is typically inside the plant, which means a firewall port must be opened to allow the connection. In the current climate of cybercrime, where port scan attacks are incessant, it only takes one open firewall port for an attacker to gain entry.

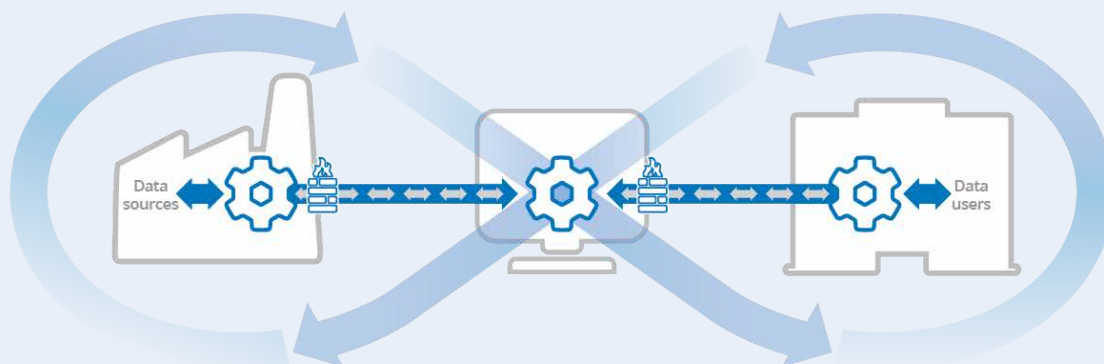
The risk goes deeper than the port or protocol. Attacks are most often directed not at the protocol itself, but at the applications that use it. An application may perfectly implement OPC UA yet be vulnerable to flaws in OpenSSL, or have a buffer overrun in a string processing function executed on incoming data. No application is free of bugs. Every open incoming port is an opportunity for an attacker to probe for exploitable flaws.

Cogent DataHub tunnel/mirroring reverses the traditional client-server relationship. The OT-side DataHub instance initiates an outbound TCP connection to the DMZ-hosted instance. No inbound port needs to be opened on the OT network firewall. The result is zero added attack surface on the OT side.

Control zone (OT)

DMZ

Corporate zone (IT)



This design can be extended across the full DMZ architecture. Outbound-only connections are configured from the OT side to the DMZ, as well as from the IT side to the DMZ. The potential attack surface moves to just the DMZ, which can then be hardened to manage the security risk. All connections support SSL encryption with 256-bit ciphers.

MQTT also supports outbound connections—each MQTT client connects outbound to a broker, keeping inbound ports closed. This is useful for simple, single-hop remote data architectures. However, as discussed below, MQTT's limitations become significant when data must cross a DMZ or traverse multiple network hops.

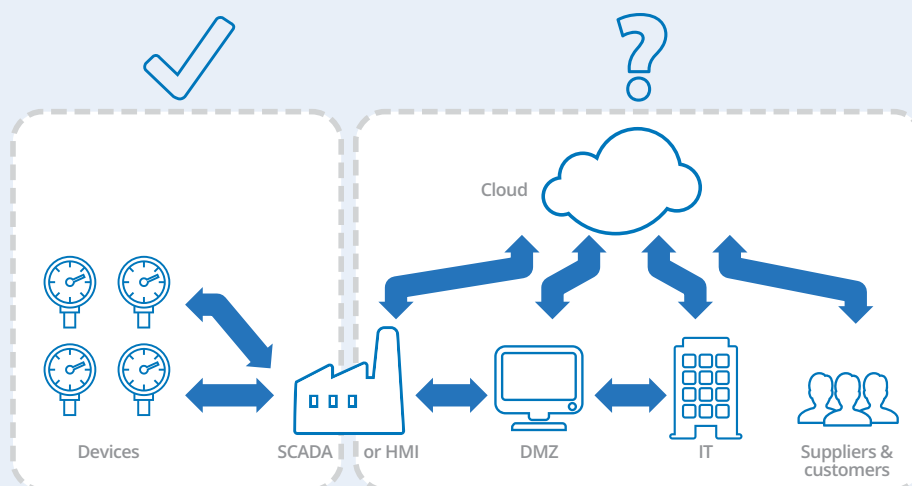
Why MQTT and OPC UA fall short at the DMZ

Implementing DMZ support is problematic for both OPC UA and MQTT. Getting data out of a plant and to a user through a DMZ typically requires two or more servers, chained

together in a daisy chain. This is where both protocols encounter fundamental limitations.

In our experience, OPC UA is simply too complex to reproduce well in a daisy chain. Information will be lost in the first hop. Attempts to daisy chain some aspects of the OPC UA protocol would result in synchronous multi-hop interactions that would be fragile on all but the most reliable networks, and would result in high latencies. Nor would OPC UA chains provide access to the data at each node in the chain.

MQTT can be chained, but it requires each node in the chain to be aware that it is part of the chain, and to be individually configured. More critically, MQTT's Quality of Service (QoS) guarantees cannot propagate through the chain. MQTT offers three QoS levels — 0 (at most once), 1 (at least once), and 2 (exactly once) — but all three were designed for single-hop connections between a client and a broker. In

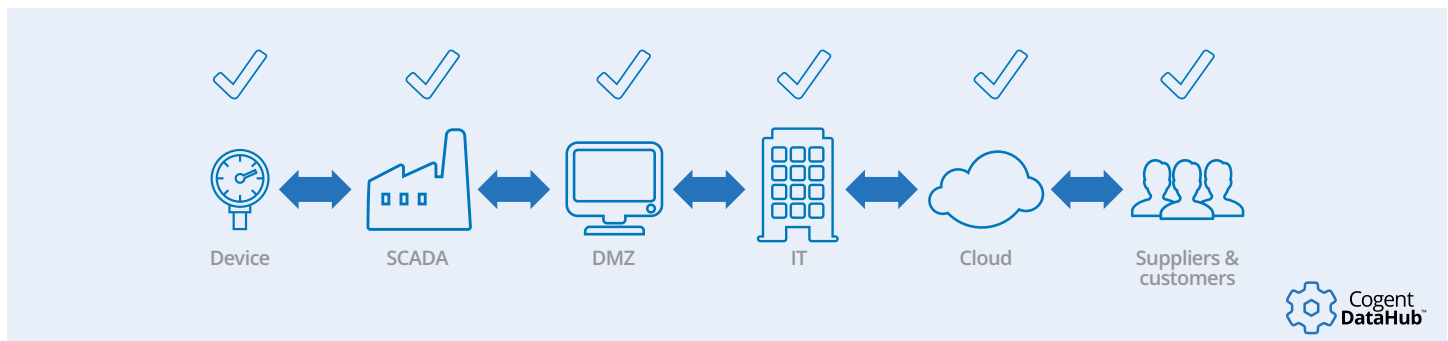


a multi-hop backbone, these guarantees do not hold. See [Which Quality of Service \(QoS\) is Right for IIoT?](#) for more details on this topic.

QoS 0 can lose messages entirely. This rules out MQTT Sparkplug, which only offers QoS 0. QoS levels 1 and 2 build up queues under load that add latency and can eventually cause the system to fail. Under heavy data rates, the system does not degrade gracefully — it falls off a cliff. And none of these levels propagate past the first broker, so a network break between the OT side and the DMZ would not be communicated reliably down the chain. Data users at the

end will not realize that their data is stale, and could make errors in judgment or wrong decisions.

Cogent DataHub tunnel/mirroring solves this problem by mirroring the full data set at each node in the chain. Each DataHub instance provides access to the data, both for qualified clients and for the next node in the chain. It guarantees consistency, so that any client or intermediate point in the chain will be consistent with the original source. If the network connection to the DMZ drops, the data quality changes to Disconnected in the DMZ-hosted DataHub instance, as well as all other downstream instances. All



data users get the notification immediately. When the connection is restored, data quality changes back to Good right away for every user at every link in the chain.

Data Diode support

For environments where even one inbound data packet is unacceptable, a data diode provides the strongest possible isolation. Like its namesake in electronics, a data diode permits data to flow in only one direction. There are no inbound packets to inspect or filter because none are ever delivered.

Data diodes may replace or complement DMZ architectures within the Purdue Model. They can be deployed at the Level 3/3.5 boundary to enforce strictly unidirectional data flow from OT to IT, or at any other boundary where one-way flow is required.

However, the one-way constraint disrupts virtually every standard industrial protocol. Both OPC UA and MQTT are built around two-way messaging — they require

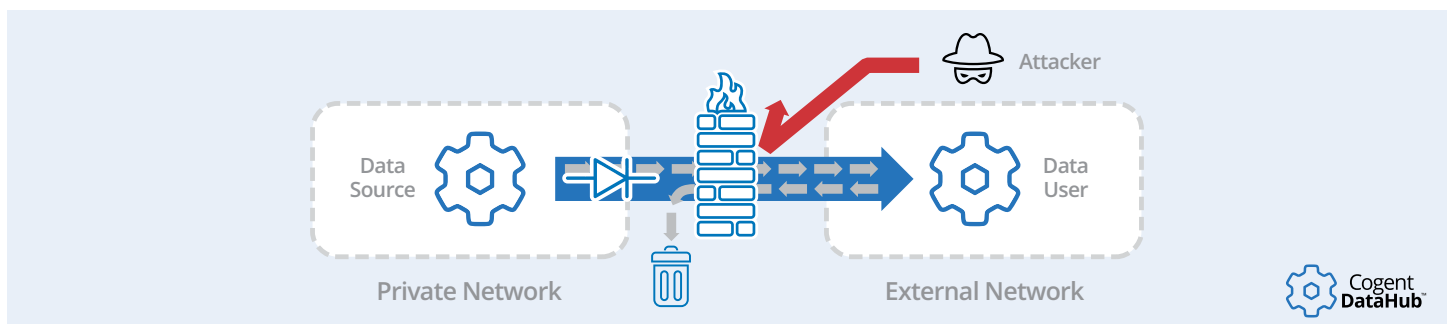
acknowledgments, subscriptions, or handshake exchanges that a true data diode blocks outright.

OPC UA's Pub/Sub model can support one-way transmission over UDP, but UDP offers no guarantees of delivery, ordering, or completeness. Packets may be dropped or arrive out of sequence, which is unacceptable in critical process environments.

MQTT requires bidirectional flows for session management and quality of service monitoring. It cannot function across a hardware data diode without encapsulation.

Cogent DataHub tunnel/mirroring supports data diode operation in two ways:

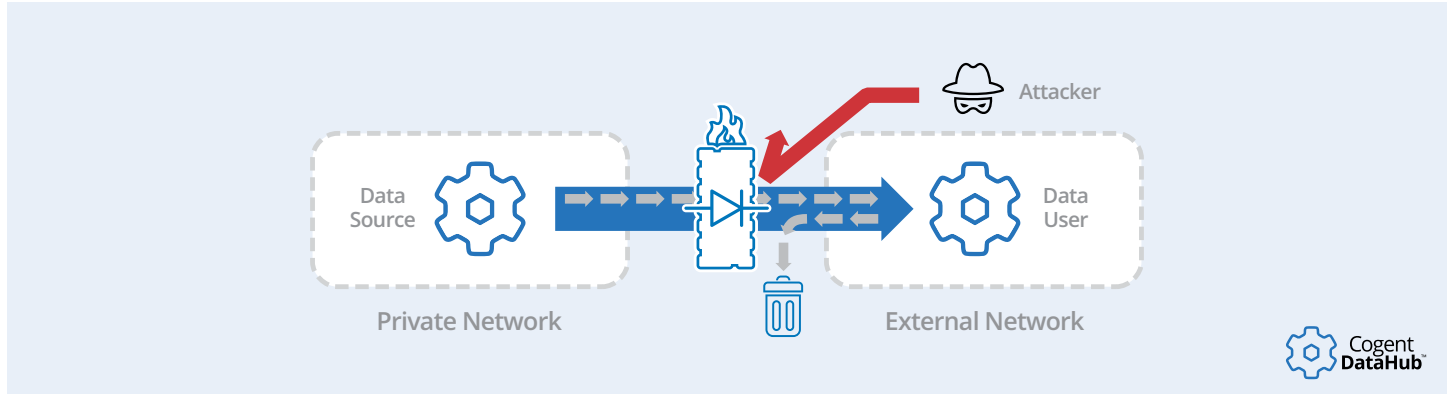
Software emulation. DataHub data diode mode makes a tunnel connection behave like a hardware data diode by discarding all incoming application data at the source without processing it. Since the data is discarded without being processed, there is no chance of application flaws being exploited by malicious packets. TCP control packets



and SSL protocol packets are still processed, so SSL connections are supported. Data flow over this connection is strictly unidirectional.

Hardware data diode support. DataHub tunnel/mirroring

can also work with most hardware data diodes. The hardware diode blocks all incoming TCP packets entirely — they are simply not delivered. Each DataHub instance provides connectivity to multiple industrial protocols (OPC UA, OPC DA, MQTT, Modbus) in a unified namespace on both sides



of the diode. The one limitation is that SSL is not available when connecting through a hardware data diode, since the SSL handshake requires bidirectional communication.

Some hardware data diodes support only UDP which, as mentioned, offers no guarantees of delivery, ordering, or completeness. To overcome these limitations, DataHub software implements a forward-only reliability layer consisting of pacing, interleaving redundant packet copies, and receiver-side reordering with bounded hold locks.

Whether by software or for hardware data diodes, the DataHub tunnel/mirror architecture encapsulates the source protocol within a unidirectional transport. On the receiving side, a mirrored DataHub instance can reconstruct the original protocol semantics for consuming applications, or provide it in an alternative protocol. The most recent value of each data point is always accurate even if intermediate updates are lost, ensuring guaranteed data consistency.

Within the Purdue Model, this means that Level 4 and Level 5 consumers can receive real-time OT data through a data diode without any protocol-level path back into the OT zone. The physical process is protected by the strongest isolation available, while analytics, historians, and enterprise systems still receive the data they need—accurate and up-to-date by the millisecond.

Summary: What the architecture requires

The Purdue Model has endured for more than three decades because its core principle — functional separation between operational levels — remains sound. The addition of the DMZ at Level 3.5, outbound-only connection requirements, and data diode support reflect the evolving

threat landscape, but the architecture itself has proven remarkably adaptable.

What has changed is the role of the DMZ layer. It is no longer enough to simply place a firewall between OT and IT and open up a few ports for data access. The DMZ must support outbound-only connections that keep every inbound firewall port closed. It must guarantee data consistency through multiple network hops, something that neither MQTT nor OPC UA can provide reliably. And for the most sensitive environments, it must support data diode operation—hardware or software—without sacrificing protocol compatibility.

Software that can operate across multiple Purdue levels, support every major industrial protocol, maintain guaranteed data consistency from source to user, and enforce outbound-only or unidirectional data flow is not optional. It is what the architecture requires.

About Skkynet

Skkynet is a global leader in real-time software and services that allow companies to securely acquire, monitor, control, visualize, network and consolidate live process data in-plant or in the cloud. DataHub™, and DataHub™ for Azure, enable secure, real-time data connectivity for industrial automation, Industrial IoT, and Industrie 4.0. Visit [skkynet.com](https://www.skkynet.com) for more about the company and [cogentdatahub.com](https://www.cogentdatahub.com) for more about Cogent DataHub.

Skkynet™, DataHub™, Cogent DataHub™, the Skkynet and DataHub logos are either registered trademarks or trademarks used under license by the Skkynet group of companies ("Skkynet") in the USA and elsewhere. All other trademarks, service marks, trade names, product names and logos are the property of their respective owners.