What makes an ideal IIoT Protocol?

WHITE PAPER

A good IIoT protocol is the basis for effective IIoT data communication. Without a secure, robust IIoT protocol, data can be late, missing, inconsistent, or dangerously incorrect, leading to costly errors and wasted time.

When IIoT was introduced, companies turned to familiar, well-tested data communication and messaging protocols such as MQTT, AMQP, REST and OPC UA for an IIoT protocol. Valid as these may be for their designed purposes, they were never intended to support IIoT data communication. Thus, when evaluated according to criteria for a robust, secure Industrial IoT implementation, they all come up somewhat short.

Skkynet's software and services are designed for the IIoT, and meet all of the criteria for effective data communication. Here we provide a comparison report on how well MQTT, AMQP, REST, OPC UA, and Skkynet's own DHTP (DataHub Transfer Protocol) meet the criteria summarized in the above table for an ideal IIoT protocol. Each of the criteria enumerated above is explained in further detail in subsequent sections.

CRITERIA	MQTT	AMQP	REST	OPC UA	DHTP
Closed Firewalls	✓	✓	✓	✗	✓
Data Diode Compatible	✗	✗	✗	✗	✓
DMZ Compatible	~	~	✗	✗	✓
Low Bandwidth & Low Latency	✓	✓	✗	◐	✓
Ability to Scale	✓	✓	✗	◐	✓
Real-Time	◐	◐	✗	✓	✓
Interoperable Data Format	◐	✗	✗	✓	✓
Intelligent Overload Handling	✗	✗	✗	✗	✓
Can Daisy Chain Servers	~	~	✗	✗	✓
Propagation of Failure Notification	~	~	✗	✗	✓
Quality of Service	~	~	✗	◐	✓

✓ YES
~ FRAGILE
◐ PARTIAL
✗ NO

Each of the criteria enumerated above is explained in further detail in subsequent sections.

Closed Firewalls

MQTT	AMQP	REST	OPC UA	DHTP
✓	✓	✓	✗	✓

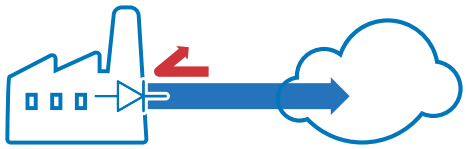
Keeps all inbound firewall ports closed for both data sources and data users.

Keeping all inbound firewall ports closed at the plant resolves many security issues for Industrial IoT. MQTT, AMQP, REST and DHTP meet this criterion. OPC UA does not because it has a client/server architecture, which requires at least one firewall port be open on the server side (typically the plant) to allow for incoming client connections. This is an unacceptable risk for most industrial systems. Skkynet's DataHub software connects locally to servers and clients in the plant, and makes outbound connections via DHTP to DataHub service for Azure or another DataHub running on a DMZ computer. This outbound connection keeps all inbound firewall ports closed and hides the plant from the outside world.

Data Diode Compatible

MQTT	AMQP	REST	OPC UA	DHTP
✗	✗	✗	✗	✓

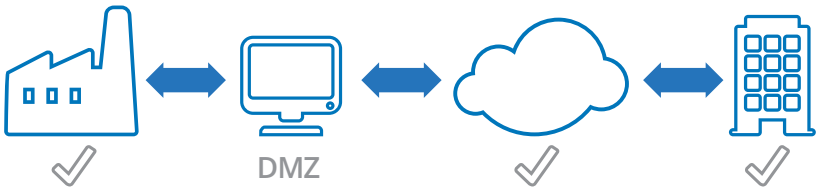
Hardware or software data diode mode blocks all incoming TCP control and application data.



A hardware data diode blocks all attacks because all inbound TCP packets are simply not delivered. As an alternative, software emulation of a data diode discards all incoming TCP control packets, SSL protocol packets, and any other application data without being processed. MQTT, AMQP, OPC UA, and REST protocols are unable to connect through a hardware data diode, and cannot provide software emulation of one. DHTP can connect through a hardware data diode, and also supports data diode software emulation.

DMZ Compatible	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	✗	✓

Network segmentation using a DMZ is critical for secure access to operations data.



Connecting an OT system through a DMZ requires propagation of failure notification, guaranteed consistency of data, and daisy chaining servers. These are explained in more detail below. As mentioned, MQTT and AMQP are unreliable for these requirements, making them a poor choice for connecting through a DMZ. OPC UA is even less suitable. DHTP mirrors the full data set through a DMZ, handling failure notifications and maintaining data consistency. It is the best option for communicating between networks that are segmented with a DMZ.

Low Bandwidth & Low Latency	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	◐	✓

Consumes minimal bandwidth, while functioning with the lowest possible latency.

One goal of any industrial communication or IIoT protocol is to consume as little bandwidth as possible, and function with the lowest possible latency. MQTT and AMQP do this well. REST does not, because every transaction includes all of the socket set-up time and communication overhead. OPC-UA is partial, because it uses a smart polling mechanism that trades bandwidth for latency. Skkynet software and services maintain a connection and transmit only the data via DHTP, consuming very little bandwidth, at very low latencies.

Ability to Scale	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	◐	✓

Can support hundreds or thousands of interconnected data sources and users.

An important aspect of the Internet of Things is the vision of connecting hundreds, thousands, and even millions of things via the Internet, and providing access to the data from any single thing, or groups of things to any number of clients. Event-driven protocols like MQTT and AMQP allow for this kind of scaling up, while REST's polling model prevents it. OPC UA is also event-driven, and so theoretically can scale up, but its underlying polling model does not allow for very large numbers of simultaneous connections DHTP abstracts the data from the protocol across the connection, and also implements an event-driven model, which allows it to scale up well.

Real-Time	MQTT	AMQP	REST	OPC UA	DHTP
	◐	◐	✗	✓	✓

Adds virtually no latency to the data transmission.

Any kind of remote HMI or supervisory control system is much more effective when functioning in at least near-real time. Propagation delays of one or more seconds may be tolerable under certain conditions or for certain use cases, but they are not ideal. AMQP and MQTT offer real-time behavior only if they are not operating with a delivery guarantee. That is, if you choose the "guaranteed delivery" quality of service then a slow connection will fall further and further behind real-time. By contrast, DHTP guarantees consistency, not individual packet delivery, and can sustain that guarantee in real time on a slow connection. REST simply has too much connection overhead to allow real-time performance in most circumstances. OPC UA, being an industrial protocol, meets this criterion well.

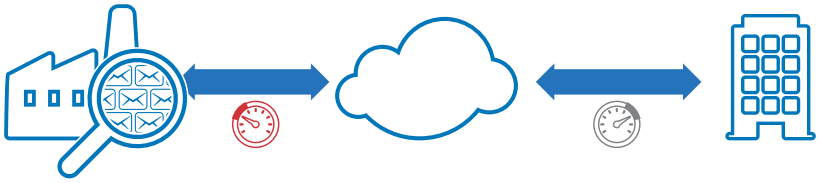
Interoperable Data Format	MQTT	AMQP	REST	OPC UA	DHTP
	◐	✗	✗	✓	✓

Encodes the data so that clients and servers do not need to know each other's protocols.

A well-defined data format is essential for interoperability, allowing any data source to communicate seamlessly with any data user. Interoperability was the primary driving force behind the original OPC protocols, and is fully supported by the OPC UA data format. Any Industrial IoT software or service should support at least one, if not multiple interoperable data formats. Skkynet's DataHub software and ETK support several, and allow for real-time interchange between them and DHTP. Regular MQTT, AMQP and REST do not support interoperability between servers and clients because they do not define the data format, only the message envelope format. MQTT Sparkplug was introduced to address this problem, and does support an interoperable data format.

Intelligent Overload Handling	MQTT	AMQP	REST	OPC UA	DHTP
	✗	✗	✗	✗	✓

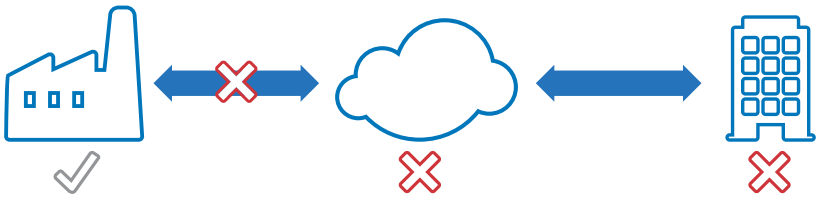
A messaging broker responds appropriately when a data user is unable to keep up with the incoming data rate.



Overload handling refers to how the broker responds when a client is unable to keep up with the incoming data rate, or when the server is unable to keep up with the incoming data rate from the client. MQTT and AMQP respond in one of two ways. Either they block, effectively becoming inoperative and blocking all clients. Or they drop new data in favor of old data, which leads to inconsistency between client and server. REST saturates its web server and becomes unresponsive. OPC UA attempts to drop old data in favor of new data, but consumes massive amounts of CPU resources to do so. When needed, Skkynet's DataHub software can drop old data efficiently, and using DHTP, it guarantee consistency between client and server even over multiple hops. Data coming from or going to overloaded clients remains consistent, and all other clients are unaffected.

Propagation of Failure Notification	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	✗	✓

Each client application knows with certainty if and when a connection anywhere along the data path has been lost, and when it recovers.



Most protocols do not provide failure notification information from within the protocol itself, but rather rely on clients to identify that a socket connection is lost. This mechanism does not propagate when there is more than one hop in the communication chain. Some protocols (such as MQTT) use a "last will and testament" that is application-specific and thus not portable, and which is only good for one connection in the chain. Clients getting data from multiple sources would need to be specifically configured to know which "last will" message is associated with which data source. In MQTT, AMQP, REST and OPC UA alike, the protocol assumes that the client will know how many hops the data is traversing, and that the client will attempt to monitor the health of all hops. That is exceptionally fragile, since knowledge about the data routing must be encoded in the client. In general, this cannot be made reliable. DHTP propagates not only the data itself, but information about the quality of the connection. Each node is fully aware of the quality of the data, and passes that information along to the next node or client.

Quality of Service	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	◐	✓

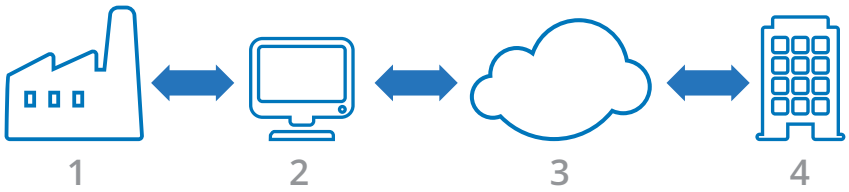
Guarantees consistency of data, preserved through multiple hops.

An important goal of the IIoT is to provide a consistent picture of the industrial data set, whether for archival, monitoring, or supervisory control. MQTT's ability to guarantee consistency of data is fragile because its Quality of Service options only apply to a single hop in the data chain. And within that single hop, delivery can be guaranteed only at the expense of losing real-time performance. Real-time performance can be preserved, but only by dropping messages and allowing

data to become inconsistent between client and server. AMQP's ability to guarantee consistency of data is fragile because like MQTT it only applies to a single hop in the chain. Additionally, its delivery guarantee blocks when the client cannot keep up with the server and becomes saturated. REST provides no Quality of Service option. OPC UA does guarantee consistency, but only for the first connection, not over multiple hops. DHTP guarantees consistency, and the guarantee is preserved through any number of hops.

Can Daisy Chain Servers	MQTT	AMQP	REST	OPC UA	DHTP
	✓	✓	✗	✗	✓

Brokers can connect to other brokers to support a wide range of collection and distribution architectures.



The requirements of the IIoT take it beyond the basic client-to-server architecture of traditional industrial applications. To get data out of a plant and into another plant, corporate office, web page or client location, often through a DMZ or cloud server, typically requires two or more servers, chained together. The OPC UA protocol is simply too complex to reproduce 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. REST servers could in theory be daisy chained, but would be synchronous, and not provide access to the data at each node in the chain. MQTT and AMQP 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. The QoS guarantees in MQTT and AMQP cannot propagate through the chain, so daisy chaining makes data at the ends unreliable. Skkynet's DataHub software supports daisy-chained servers because DHTP allows each DataHub instance to mirror the full data set at each node, and provide access to that data both to qualified clients, as well as the next node in the chain. The DHTP QoS guarantee states that any client or intermediate point in the chain will be consistent with the original source, even if some events must be dropped to accommodate limited bandwidth.

In Conclusion

Far from exhaustive, this overview of effective IIoT data communication provides an introduction to the subject, and attempts to highlight some of the key concepts, through sharing what we have found to be essential criteria for evaluating some of the protocols currently on offer. Because none of MQTT, AMQP, REST, or OPC UA were designed specifically for use in Industrial IoT, it is not surprising that they do not fulfill these criteria. DHTP, on the other hand, was created specifically to meet the needs of effective industrial and IIoT data communication, making it an ideal choice for an IIoT protocol.

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™, DataHub™ for Azure, and Embedded Toolkit (ETK) software enable secure, real-time data connectivity for industrial automation, Industrial IoT, and Industrie 4.0. Visit **skkynet.com** for more about the company and **cogentdatahub.com** for more about DataHub.